NONCONVEX PROGRAMMING WITH APPLICATIONS TO PRODUCTION AND LOCATION PROBLEMS

A THESIS

Presented to

The Faculty of the Division of Graduate

Studies and Research

By

Harish Vaish

In Partial Fulfillment

of the Requirements for the Degree

Doctor of Philosophy

in the School of Industrial and Systems Engineering

Georgia Institute of Technology

December, 1974

NONCONVEX PROGRAMMING WITH APPLICATIONS TO

PRODUCTION AND LOCATION PROBLEMS

Approved:

C. ~~M.~~ Shetty, Chairman

J. J. Goode

J. J. Jarvis

L. A. Johnson

Date approved by Chairman: Nov. 24, 1974

ACKNOWLEDGMENTS

I am unable to fully express my gratitude and indebtedness to Dr. C. M. Shetty. Without his deep insight, boundless patience, infectious enthusiasm, mature guidance and warm friendship, this work would never have seen the light of day.

I wish to thank Dr. Jamie Goode, Dr. John Jarvis, Dr. L. A. Johnson and Dr. D. E. Fyffe for serving on my dissertation committee.

The counsel and financial help provided by Dr. R. N. Lehrer and Dr. W. W. Hines are greatly appreciated.

Finally, I thank Gita and Titu for their understanding and consideration.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF ILLUSTRATIONS

SUMMARY

The theory of generalized polars as developed by Balas, Burdet, and others in the context of integer and nonconvex quadratic programming problems is used to develop a cutting plane algorithm for the Bilinear Programming Problem (BLP). It is shown that the cuts generated are deeper than those generated by Konno. The algorithm converges infinitely to a global optimal solution. It is also proved that an ε-optimal solution is obtained in a finite number of steps. Some computational results are provided.

A second algorithm, based on an inductive construction of a sequence of polytopes, is developed for the BLP. This algorithm is shown to converge to a global optimum in a finite number of steps.

CHAPTER I

INTRODUCTION AND LITERATURE SURVEY

1.1 Introduction

The area of mathematical programming has attracted researchers from different disciplines ever since the development of the simplex method for linear programming and the capability of solving realistic problems on electronic computers. One of the important objectives has been to develop a comparable problem-solving capability for models more complex than those satisfying the restrictive assumptions of linear programming. Such attempts in the areas of convex and integer programming were not quite as successful as in the case of linear programming for the most general class of problems. However, relatively effective algorithms in these areas have been constructed for solving large problems with special structures by taking advantage of the special properties of the problem. As it turns out, a number of practical problems do have special structures. Hence, so long as major methodological breakthroughs are not achieved, it appears that research in this direction will become more and more important.

This is all the more true in the case of nonconvex programming. In contrast to the convex case, research on nonconvex programming problems is still in the preliminary stages. There does not exist any effective algorithm for the most general nonconvex problem. Solution procedures have been proposed for special cases, but even here several deficiencies still exist. Firstly, not many of these algorithms have a proof of convergence. In some cases, counterexamples have been published to demonstrate lack of convergence. Secondly, it does not appear easy to program these algorithms for implementation on computers. Finally, since computational results have not been widely reported, it is hard to conjecture what size of problems can be solved within the constraints of time and available high speed memory.

The specialized problem that we will be studying is the Bilinear Programming Problem, which is defined in the next section. The objectives of this study are:

1. To develop a cutting plane algorithm for solving the Bilinear Programming Problem such that the cuts generated are deeper than those proposed in the literature. It is expected that this algorithm will be computationally feasible. We will prove infinite convergence of the algorithm. We will also develop an algorithm which will converge in a finite number of steps to an ε-optimal solution.

2. To develop a second algorithm for solving the Bilinear Problem based on an inductive construction of a special polytype. This algorithm will be shown to converge in a finite number of steps. However, only smaller problems can be solved by this method since it requires a large data storage capability.

3. To gain some computational experience with the cutting plane algorithm so as to compare with the limited results available, to get a feel for the rate of convergence and for the largest problem that can be solved in a reasonable amount of time.

The organization of this study is as follows. In the remainder of this chapter, we will state the problem to be studied and the techniques available in the literature for solving the Bilinear and closely related problems. Chapter II contains a study of the possible areas of application of the Bilinear Problem so as to highlight its importance. In Chapter III we discuss the nature of the objective function of the Bilinear Problem and present an algorithm for determining a good, feasible starting point, which is used heavily later on. In Chapter IV we develop the cutting plane algorithm, and in Chapter V the algorithm based on an inductive construction of a polytope. In Chapter VI, we discuss methods for determining a lower bound for the objective function and show how it can be useful to terminate the algorithm quicker than without it. In Chapter VII we summarize our computational results based on the performance of a computer code that we have developed on a variety of test problems, state our final conclusions and indicate areas for further research.

The basic definitions and notation used in this study are standard, see for example [22] and [29]. When we write $X \subset Y$, X could be equal to Y, unless stated otherwise. We will be using a number of results which have been derived in the literature of this area. We will simply state them without proof and cite an appropriate reference. In the rest of this chapter, uncommon terms will be heuristically explained. They will be defined precisely later on when they are used.

1.2 Problem Statement

The Bilinear Programming Problem is stated mathematially as:

$$\text{(BLP 1)}\quad \text{Minimize}\quad \phi(x,y) = c^t x + d^t y + x^t Cy$$
$$\text{Subject to}\quad Ex \le e,\ x \ge 0$$
$$Fy \le f,\ y \ge 0$$

where $x \in R^m$, $y \in R^n$ are variable vectors, $c \in R^m$, $d \in R^n$, $e \in R^{k_1}$, $f \in R^{k_2}$ are given vectors, and $C \in R^{m\times n}$, $E \in R^{k_1\times m}$, $F \in R^{k_2\times n}$ are given matrices. For simplicity in presentation we will define the following sets:

$$X_0 = \{x\epsilon R^m | Ex\le e,\ x\ge 0\}$$

$$Y_0 = \{y\epsilon R^n | Fy\le f,\ y\ge 0\}$$

We will assume that X_0 and Y_0 are nonempty polytopes, i.e. bounded polyhedral sets. Since a system of equations can always be expressed as a system of inequalities, and since an unbounded constraint set can be bounded by introducing appropriate constraints, there is no loss in generality in assuming the nature of X_0 and Y_0.

It is to be noted that the constraints are separable in the variables x and y, but the objective function is not. However, for a fixed x (or y) ϕ becomes linear in y (or x). This fact will play a key role later on.

1.3 Solution Procedures for BLP 1

The Bilinear Problem comes under the category of problems called Quadratic Programming Problems in which convexity assumptions have been relaxed. This relationship is shown in Section 3.2. We will, therefore, be reviewing algorithms for this class of problems in addition to those proposed for BLP 1. There exist a number of well-known algorithms for Convex Quadratic Programming problems, but these will not be stated here.

1. Cabot and Francis' Algorithm [10]

The problem is stated as

$$\text{P1:} \quad \text{Minimize} \quad f(x) = c^t x + x^t D x$$
$$\text{Subject to} \quad x \in S = \{x \mid Ax=b,\ x \geq 0\}$$

It is assumed that an extreme point of the convex set S is an optimal solution to P1. This may be so either because of some special properties of f (for example, concavity) or because an extreme point optimal solution is desired, as, for example, in a quadratic assignment problem. Murty's [35] extreme point ranking procedure is used to solve the problem. In order to do so, upper and lower bounds are developed for the optimal value of f. Let u_j be the minimum value of the objective function of the following problem:

$$\text{Minimize} \quad (d^j)^t x$$
$$\text{Subject to} \quad x \in S; \quad j = 1,\ldots,n.$$

where d^j is the j*th* column of D. A bounding linear programming problem P2 is defined:

$$\text{P2:}\quad \text{Minimize} \quad g(x) = \sum_{j=1}^{n} (c_j + u_j)x_j$$
$$\text{Subject to} \quad x \in S$$

From the definition of the u_j, it is readily seen that for any $x \in S$, $g(x) \le f(x)$. Thus, if x^0 is an optimal solution to P2, then $f_L = g(x^0)$ is a lower bound on f, a readily obtainable upper bound being $f_u = f(x^0)$.

The extreme point ranking procedure is applied to P2. If, at a certain stage, the k*th* ranked extreme point x^k is such that $g(x^k) > f_u$, then clearly the solution corresponding to f_u is optimal to P1. If $g(x^k) \le f_u$, f_L is revised and set equal to $g(x^k)$ which is an improved lower bound. If $f(x^k) < f_u$, then $f(x^k)$ is obviously an improved estimate of the upper bound. In this way, the bounds approach each other till termination occurs. A 4 × 6 transportation problem was solved in 10 seconds on CDC 3600.

2. Taha's Algorithm [45]

This is an algorithm for minimizing a concave function defined over a polyhedral set. As applied to a quadratic objective function, the algorithm is the same as the method of Cabot and Francis, the only difference being in the method used for ranking the extreme points. Ranking extreme points according to Murty's algorithm requires examination of several tableaux, and has been found to be not very efficient. Taha generates a cutting plane at each stage which cuts off only ranked extreme points. Solving a linear programming problem over the remaining feasible region automatically generates the next best extreme point of the original set, since it can be shown that extreme points created by the cutting plane cannot be optimal to the remaining problem. Computational results when the algorithm is applied to a fixed charge problem compare very favorably with those of Steinberg [44].

3. The Method of Mueller [34]

The problem is stated as:

$$\text{Maximize} \quad z(x) = c^t x + x^t F x$$

$$\text{Subject to} \quad x \in C = \{x \mid a_i x \le b_i, \ i=1,\ldots,m\}$$

where F is an *indefinite* matrix. It can be shown that the optimal solution to this problem is not an interior point of C. Moreover, if $\bar{x}$ is the optimal solution, I the set of binding constraints at $\bar{x}$ and $L = \{x \mid a_i x = b_i, \ i \in I\}$, then the function z(x) restricted to L is a concave quadratic function. It now follows that the optimal solution to the problem will be either an extreme point of C or a stationary point of the function z(x) restricted to *some* boundary L of C. The basic ideas of the method are as follows. Let us first assume that there are no stationary points of z(x) in C. Picking any arbitrary starting point in C and using gradient projection to determine the direction of movement at each stage, a sequence of optimal moves is made in the directions selected till one of several things occur. If at any stage, movement along a direction ends at an extreme point, then at the next stage all adjacent extreme points are examined. If there exists one with a higher value of z(x), we move to that extreme point and continue the search. If not, another arbitrary starting point is selected and the whole algorithm repeated. Similarly, if we end up at a boundary point which is not a stationary point of z(x) restricted to the active constraints at that point, no further direction of movement can be determined, so that another arbitrary starting point will have to be selected and the algorithm repeated. If we end up at a boundary point and z(x) is concave when restricted to the constraints active at that point, once again we have a candidate for the global optimum, but since there is no way of recognizing it, another starting point will have to be generated. In the case when the ending point is a stationary point of z(x) restricted to the active constraints, a direction of movement is defined according to a special procedure which will ensure further improvement in the objective function. The algorithm terminates when a specified time limit has elapsed. It is recognized that there is no guarantee that the global optimum will have been found. If there does exist a stationary point of z(x) in C, the only difference is in the method of selecting the starting points. Half of all the starting points used must have an objective function value greater than the function value at the stationary point. The remaining starting points are derived from the ones selected according to a prescribed rule. The reason for this is the fact that the set of all x with z(x) greater than that at the stationary point consists of at most two disjoint polygonally connected components, and both these components have to be searched. Although no proof of convergence is provided, some computational results indicate that most of the time the optimal solution was obtained. No computational times are reported, since, apparently, the objective was only to test convergence.

4. The Method of Candler and Townsley [11]

The problem considered is:

$$\text{Maximize} \quad c^t x + x^t C x$$
$$\text{Subject to} \quad Ax = b, \quad x \geq 0.$$

The first phase of solving the problem consists of finding a local maximum. Starting from an extreme point, a sequence of simplex operations is defined to move to better adjacent extreme points until either an unbounded solution is indicated, in which case the problem is solved, or a position is reached such that a move with an improvement in objective function value can be made along an edge not all the way to the adjacent extreme point but to a point in between on the edge. In this case the original problem is augmented by one constraint and two variables such that a vertex of the modified problem has a higher value of the objective function. This process of augmentation is continued till a local maximum is obtained, when no further movement can be made along any edge.

Given a local maximum at a vertex, the basic variables can be expressed in terms of the nonbasic variables at this vertex. Using these equations to eliminate the basic variables, a new objective function is obtained in terms of the nonbasic variables $\bar{x}$ only: $k + \bar{c}^t\bar{x} + \bar{x}^t\bar{C}\bar{x}$, where k is a constant. Since this is a local maximum, $\bar{c} < 0$. Now we would like to consider the question of how much the nonbasic variables $\bar{x}$ can be increased without reducing the objective function value below k. Clearly, $(\bar{c}^t+\bar{x}^t\bar{C})\bar{x} \leq 0$ for $\bar{x} \geq 0$ if $\bar{c}^t + \bar{x}^t\bar{C} \leq 0$, that is if $\bar{x}^t\bar{C} \leq |\bar{c}|$. One inequality which ensures that $\bar{x}^t\bar{C} \leq |\bar{c}|$ is not violated is derived as follows. Let $\lambda_j = \min_i\{\frac{|\bar{c}_i|}{\bar{c}_{ij}}\}$, $\bar{c}_{ij} > 0$, where $\bar{c}_{ij}$ are the elements of C. If only the component $\bar{x}_j$ is increased and all others fixed at zero, then λ_j is the maximum that $\bar{x}_j$ can be increased without violating the system $\bar{x}^t\bar{C} \leq |\bar{c}|$. If all $\bar{c}_{ij} \leq 0$, then clearly $\lambda_j = \infty$. The inequality required is then $\sum \bar{x}_j|\lambda_j \leq 1$. The cutting plane to be added to the original system is thus: $\sum \bar{x}_j|\lambda_j \geq 1$. This will cause the feasible region to be reduced. After a number of iterations, the feasible region will be depleted. However, no proof of convergence has been provided nor any computational results reported.

5. The Method of Tui [46]

The problem considered is:

$$\text{Minimize} \quad f(x)$$
$$\text{Subject to} \quad x \in D \subset R^n$$

where D is a polyhedron and f is a concave function. This method closely resembles the algorithm described in (4) above. Since f is concave, its minimum over D will be attained at an extreme point of D. It can be shown that for a concave function, a local star minimum is also a local minimum, where a local star minimum is an extreme point $\bar{x}$ at which the objective function value is not greater than that at any extreme point adjacent to $\bar{x}$. The search for a local minimum is therefore considerably simplified. A sequence of moves is made from one extreme point of D to an adjacent extreme point till a local star minimum x^0 is located. Tui assumes that precisely n edges are incident on x^0, in the degenerate case perturbation methods being used to achieve this objective. A one-dimensional search is made along each of these n rays to find the points $y^1,y^2,\ldots,y^n$ furthest from x^0 such that $f(y^i) \geq f(x^0)$. The minimum over the convex hull of the points $\{x^0,y^1,y^2,\ldots,y^n\}$ is known and is equal to $f(x^0)$. H is now a valid cutting plane, where H is the hyperplane passing through $y^1,y^2,\ldots,y^n$ and $x^0 \in \text{int}[H^-]$. Actually instead of f, Tui uses the "best concave extension of f" to obtain deeper cuts. Once again, no proof of convergence is available nor have any computational results been reported.

Tui presents a second method for solving the above problem which does not require the addition of constraints at each iteration. This can be particularly useful when the problem has a special structure, for example, a transportation problem. The basic idea of the method is to construct a polytope P such that if D is not a subset of P, an extreme point of D not in P is generated, and an enlarged polytope P' is defined from P by including the newly generated extreme point. To start off, the hyperplane H is defined as above, and the initial polytope is the convex hull of the points $\{x^0,y^1,\ldots,y^n\}$. The extreme point of D furthest from H is determined by solving the following linear programming problem:

$$\text{P3: Maximize} \quad e^t B^{-1} x$$
$$\text{Subject to} \quad x \in D$$

where e is a vector of ones and B is a matrix whose columns are the vectors $y^1,\ldots,y^n$. If the optimal value of the objective function is ≤ 1, then $D \subset P$ and f(x) is the optimal solution. If not, an extreme point x' of D not in P is found and the current global minimum is set equal to min {f(x),f(x')}. The point x' is projected along the line joining x to x' to a point $\bar{x}'$ such that $f(\bar{x}')$ is equal to the current global minimum, $\bar{x}'$ is then expressed as a linear combination of the points $y^1,\ldots,y^n$, and each y^i whose coefficient in this linear expression is nonzero is replaced, in turn, by $\bar{x}'$ and hyperplanes defined through these sets of points. Thus, at this stage, at most n hyperplanes will be defined. The problem P3 is solved with respect to each of these hyperplanes and the procedure is continued till the optimal solution is less than or equal to 1 for each hyperplane generated, which implies that $D \subset P$. Tui asserted that "it can be proved that the process must terminate in a finite number of steps." However, a counterexample was provided by Zwart [50]. We will elaborate more on this method in Chapter V.

Hu [24] has suggested some minor modifications of Tui's cutting plane algorithm to take care of degeneracy, which Tui has virtually ignored. In the case when more adjacent extreme points to the local star minimum are obtained than are necessary to uniquely define a cutting plane, Hu picks the required number and defines a cutting plane. He then examines the objective function value at the point of intersection of this plane with each of the rays defined by joining the local star minimum to the remaining extreme points. If the value is not less than the current global minimum, then clearly a valid cutting plane has been defined. If not, the plane is "moved" parallel to itself towards the local star minimum till the cutting plane becomes valid. This does not, of course, solve the problem of convergence. Hu also suggests a method for defining a concave extension to f: a plane tangent to f along each edge of D.

6. The Method of Carvajal-Moreno [33]

The problem considered is exactly the same as in (5) above: the minimization of a concave function over a polyhedron. It is first demonstrated that Tui [46] is incorrect in asserting that perturbation methods can take care of degeneracy. To overcome the problem, all the adjacent extreme points to the local star minimum are generated. These points are then projected along their respective rays exactly as in Tui's method. There could now exist more points than are necessary to uniquely define a cutting plane. If the projected adjacent extreme points to the local star minimum x^0 are x^1, x^2 and x^3, then the hyperplane passing through x^1 and x^2 is the only valid cutting plane (see Figure 1). If x^2 were at infinity, one could construct two valid cutting planes, one through (x^1,x^3) and the other through (x^2,x^3).

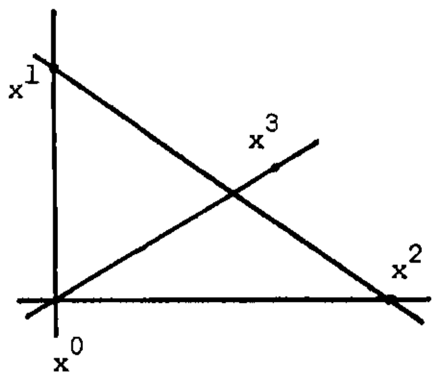


Figure 1. Example of Valid Cutting Plane

The best hyperplane will be the one which cuts off the largest portion of the feasible region. If $p^t x \geq p_0$ is the cutting plane to be defined, the following linear programming problem is used to determine the values of p and p_0:

$$\begin{aligned} &\text{Maximize} && p_0 \\ &\text{Subject to} && (x^i)^t p \geq p_0 \\ & && \sum_{i=1}^{n} (x^i)^t p \leq 1 \end{aligned}$$

where $x^1, \ldots, x^n$ are the projections of the extreme points adjacent to x^0. The last constraint ensures that p_0 is bounded and that the resulting hyperplane is a valid cutting plane. The algorithm thus consists of three phases: determination of a local star minimum, determination of adjacent extreme points, and definition of a cutting plane. No attempt is made to prove convergence, but the global minimum was located for each of the ten problems that the algorithm was tested on.

7. The Method of Zwart [49]

The problem considered is once again the minimization of a concave function over a polyhedron. The local star minimum x^0 and its adjacent extreme points are generated. Let these points be $x^1, x^2, \ldots, x^n$. Each point x^i is projected along the line joining x^0 to x^i to the furthest point e^i such that $f(e^i) \geq f(x^0)$. Let S_1 be the convex hull of the points $x^0, e^1, \ldots, e^n$, and let $C = \{x | x = x^0 + \lambda(y - x^0),\ y \in S_1,\ \lambda \geq 0\}$ be the cone generated by S_1. Since the minimum over S_1 is known, if $X \subset S_1$, the problem is solved. Otherwise, a point $x^k \in X$, $x^k \notin S_1$ is found and other cones are generated which are subsets of the parent cone, and the method continues. There are two essential differences from Tui's second method. Firstly, Zwart solves a modified problem P3 which essentially asks the question: is there a point $x^k \in X$, $x^k \notin S_1$ and $x^k \in C$? Tui did not confine his attention to points within the cone itself. A consequence of this is that the solution to Zwart's modified problem P3 may not be an extreme point of X since additional constraints are added to find the point within the cone. The second difference is that if an extreme point $x^k \notin S_1$ is found, then other cones are generated if the distance of x^k from the hyperplane passing through $e^1, \ldots, e^n$ is greater than some given ε. This ensures convergence to an ε-optimal solution, which is a solution with objective function value within ε of the global minimum. This algorithm was programmed and tested on randomly generated data. A problem with 15 variables and 30 constraints was solved in less than 10 minutes on the IBM 360/50. It is also reported that the global minimum was found early in the computations.

8. The Method of Burdet and Balas [4]

The problem considered is one of maximizing a convex quadratic function f subject to linear constraints. The problem is thus a special case of the one considered by Tui and Zwart. This algorithm is also of the cutting plane type. After a local star minimum $\bar{x}$ has been obtained, by considering the binding constraints at $\bar{x}$, precisely m edges are defined emanating from $\bar{x}$, where m is the number of nonbasic variables. Tui's cutting plane is determined by finding the point of intersection of each of these rays with the level set of f, where the level set is defined with respect to the current global minimum value of f. In this method, the point of intersection of each ray with a set larger than the level set is determined. The larger set is defined in terms of generalized polars, which we will go into in greater detail in Chapter IV and which forms the basis of the cutting plane derived therein. Since the polaroid set contains the level set, the cut generated is deeper than Tui's cut. However, as seems to be the problem with most cutting plane algorithms, no proof of convergence is available. Moreover, no computational results are available. A cutting plane algorithm has been developed in [3] for a general quadratic programming problem based on generalized polars.

9. Ritter's Method [16,38]

This is perhaps the most widely known algorithm in this area. It locates the global minimum of a general quadratic function defined over a polyhedron. It consists of three parts. The first one finds a vertex of the feasible region using the simplex procedure. Then a local minimum is determined using a characterization of the stationary points of a quadratic program. Finally, a cutting plane is developed that eliminates the local minimum without deleting the global minimum if it has not yet been found. These three steps are repeated till either the feasible region is empty or an unbounded solution is indicated or a sufficient condition for a global minimum is satisfied. For the sake of simplicity, we will describe the algorithm when the objective function is concave.

A vertex x that is a local star minimum is determined. Then x is also a local minimum. Assuming x to be nondegenerate, by a suitable transformation of variables the origin can be transferred to x and the transformed problem can be expressed as:

$$\begin{array}{ll} \text{Minimize} & f(v) = c^t v + \tfrac{1}{2} v^t E v + k \\ \text{Subject to} & Av \geq b, \quad v \geq 0 \end{array}$$

We wish to reduce the feasible region by adding a cutting plane of the kind $d^t v \geq \bar{t}$ which does not cut off the global minimum. In order to do so, a vector d and a scalar $\bar{t}$ needs to be specified. Since the origin is a local minimum, $c \geq 0$. Ritter chooses $c^t = d^t$ and calculates $\bar{t}$ by solving the following problem:

$$\begin{array}{ll} \text{Minimize} & c^t v + \tfrac{1}{2} v^t E v \\ \text{Subject to} & d^t v = t \\ & v \geq 0 \text{ for all } t \end{array}$$

The Kuhn-Tucker conditions for this problem are:

$$u = c + Ev + d\xi \tag{1}$$

$$t = d^t v \tag{2}$$

$$0 = v^t u \tag{3}$$

$$u \geq 0, \quad v \geq 0 \tag{4}$$

If $d^t = c^t$, then the above equations reduce to

$$u = c\sigma + Ev \tag{5}$$

$$t = c^t v \tag{6}$$

$$0 = v^t u \tag{7}$$

$$u \geq 0, \quad v \geq 0, \quad \sigma \geq 0 \tag{8}$$

where $\sigma = 1 + \xi$. Also, the objective function simplifies to $t - \frac{1}{2} t\sigma$. Since the last system of equations is homogenous in t, we can set $t = 1$. Thus the problem that needs to be solved is:

$$\begin{array}{ll} \text{Maximize} & \sigma \\ \text{Subject to} & u = c\sigma + Ev \\ & 1 = c^t v \\ & 0 = v^t u \\ & u, v, \sigma \geq 0 \end{array}$$

Since the original problem is concave, its solution will be at a vertex of $\{v | d^t v = 1,\ v \geq 0\}$. Hence the solution to the above problem will have precisely one component of v that will be positive and will satisfy the condition: $d^t v = 1$. By simple enumeration we can find values of u and v that maximize σ. Let these be u^*, v^*, σ^*. If f_{min} is the current global minimum, we want to find the largest value of t so that the corresponding solution tv^* will have the following property which ensures that the global minimum is not cut off: $c^t(tv^*) + \frac{1}{2}(tv^*)^t E(tv^*) + k \geq f_{min}$. We can also increase t so long as the corresponding solution tv^* remains feasible to the remaining constraints $Av \geq b$ which were ignored. The larger of the two t values is set equal to $\bar{t}$. It is clear that if the second value of t is larger, then the current global minimum is tv^*. If this t is infinite, the solution is unbounded. If the first t is infinite, then the global minimum has been found.

From Equations (1), (2), and (3), and knowing that d = c, we can see that $t = (v^t Ev)/(1+\xi)$. Ritter erroneously concluded that ξ was bounded from above so that there is a lower bound on t strictly greater than 0. Zwart [50] provided an example where ξ tends to infinity. Thus Ritter's algorithm may fail to converge. There are no computational results available on the performance of this algorithm as applied to a quadratic objective function. However, the conclusion reached by Zwart [51] is that Tui's cutting plane algorithm does poorly for problems with as little as 10 variables and 15 constraints, and since Tui's cut dominates that of Ritter for a concave objective function, it is expected that Ritter's algorithm will not do very well either.

10. The Method of Konno [26]

The problem considered is the one stated in Section 1.2. A cutting plane is developed based on Ritter's algorithm. A cut of the form $g^t x \geq \bar{\sigma}$ is desired, where the vector g and the scalar $\bar{\sigma}$ needs to be specified. First, a local star minimum is determined with the help of the following strategy. Let x^1 be an arbitrary solution to the constraints in x. By fixing $x = x^1$, BLP 1 reduces to a linear program in y, which can be solved to obtain an optimal solution y^1. By fixing $y = y^1$ and solving the resulting linear program in x, an optimal solution x^2 is obtained. This process is continued till a point $(\bar{x},\bar{y})$ is obtained such that $\bar{y}$ is the solution obtained with x fixed at $\bar{x}$ and vice versa. Then $(\bar{x},\bar{y})$ is a local star minimum. By a transformation of variables, the origin of the new coordinate system can be defined at the point $(\bar{x},\bar{y})$ in terms of the old system. In order to define the cut, the following problem is considered:

$$
\begin{aligned}
\text{P4:}\quad & \text{Minimize} \quad && \phi(x,y) = c^t x + d^t y + x^t C y \\
& \text{Subject to} \quad && g^t x \leq \sigma, \quad x \geq 0 \\
& && y \in Y_0 = \{y \mid Fy \leq f,\ y \geq 0\}
\end{aligned}
$$

It can be shown that at the local star minimum, $c \geq 0$ and $d \geq 0$. The vector g is first defined as follows:

$$g_i = c_i \text{ if } c_i > 0$$

$$g_i = g_0 \text{ if } c_i = 0,$$

where g_0 is some positive constant. One can see that for a fixed y, problem P4 reduces to a linear program in x with only one constraint, $g^t x \leq \sigma$, so that its optimal solution will have at most one variable x_i positive and of the form σ/g_i. As the hyperplane $g^t x = \sigma$ is translated parallel to itself by varying σ, the intercepts made on the rays emanating from the origin are σ/g_i from the origin for ray i. For each such point, in order to find the minimum value of $\phi(x,y)$ for all $y \in Y_0$, we need to solve the following parametric problem in σ:

$$\begin{aligned} &\text{Min}[d^t + \sigma/g_i(c^i)^t]y + c_i\sigma/g_i = \bar{z}_i \\ &\text{Subject to } \; y \in Y_0, \;\; \sigma > 0 \end{aligned}$$

where c^i is the *ith* row of C. For each i, let $\bar{\sigma}_i$ be the largest value of σ such that $\bar{z}_i + \phi_0 \geq \phi_{min}$, where ϕ_0 is the objective function value at the local star minimum and ϕ_{min} is the current global minimum. This inequality will ensure that the global minimum of the problem is not cut off. Having found the furthest one can move along each ray a valid cutting plane will be defined by moving the minimum of the permissible distances along the rays, that is, by setting $\bar{\sigma} = \min\{\bar{\sigma}_1, \bar{\sigma}_2, \ldots, \bar{\sigma}_n\}$. A cut can be similarly defined with respect to the y variables. One can, of course, define a cut to include both x and y variables, but this would destroy the separable structure of the problem. Konno is able to prove convergence to an ε-optimal solution in a very devious manner. We will comment more on this in Chapter IV. No computational results are reported.

11. The Method of Gallo and Ülkücü [18]

The problem considered is the one stated in Section 1.2. However, they work with the following problem which is equivalent because of duality theory:

$$\begin{array}{ll} \text{Minimize} & (c^t x + \text{Min } f^t u) \\ \text{Subject to} & Ex \le e, \quad x \ge 0 \\ & F^t u \ge -d - C^t x, \quad u \ge 0 \end{array}$$

The following sets are defined:

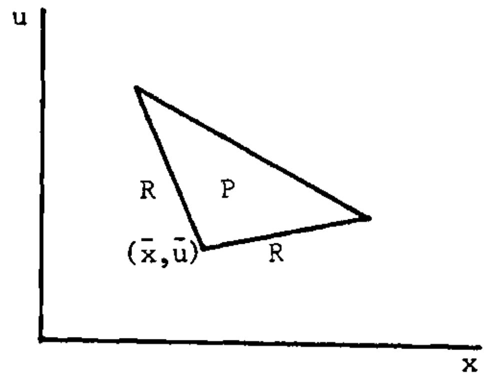
$$P = \{(x,u) \mid Ex \le e;\ F^t u \ge -d - C^t x,\ (x,u) \ge 0\}$$

$$R = \{(x,u) \mid (x,u) \in P;\ f^t u \le f^t u' \text{ for all } (x,u') \in P\}$$

The optimization problem can now be restated as:

$$\text{Minimize}\{c^t x + f^t u \mid (x,u) \in R\}$$

This problem has a linear objective function, but its set of feasible solutions is not, in general, a convex set. The following example demonstrates these conditions, where R is nonconvex.



We will discuss the method presented for solving this problem in Chapter V. The authors "prove" that their algorithm converges in a finite number of steps. However, applying this algorithm to the numerical example of Zwart [50], we have shown that it may not converge (see Chapter V). We have pointed this out to the authors, but have not received any response.

In addition to the methods outlined above, there exists an algorithm due to Falk and Soland [17] for a separable nonconvex programming problem. This has been specialized in [42] to solve a minimization problem with a separable concave objective function and linear constraints. We will not be discussing this algorithm since it is not applicable to the Bilinear problem. Also Mylander [36] has developed a method for solving a nonconvex quadratic programming problem which consists of enumerating all the Kuhn-Tucker points of the problem. We have not been able to get a copy of this report.

1.4 Summary

We have reviewed a number of algorithms in this chapter, many of which can be used to solve the Bilinear Problem. Apart from complete enumeration of all the extreme points or all the Kuhn-Tucker points, the available algorithms can be classified as follows:

1. Hueristic search procedure.
2. Ranking the extreme points.
3. Cutting plane methods.
4. Polytope generation methods.

The extreme point ranking methods are finitely convergent. Of the several algorithms of the third and fourth categories, only one from each has been shown to converge to an ε-optimal solution in the literature. The remaining may either not converge or do not have a proof of finite convergence. Only four papers report some computational results; however, even these are somewhat inadequate.

CHAPTER II

APPLICATIONS

2.1 Introduction

The Bilinear Problem as stated in Section 1.2 has a special structure. Its constraints are separable in x and y, and its objective function contains linear and cross product terms only. For a fixed x (or y), the Bilinear Problem reduces to a linear programming problem. A number of practical problems can be formulated as a Bilinear Programming Problem. For this reason, it is important that efficient computational techniques be available to solve such a problem, and that we fully understand its properties. Thus, in spite of its special structure, it is an important enough problem and worth investigating. In this chapter, we will show how certain problems can be formulated as a Bilinear Programming Problem. Some applications of Bilinear Programming are discussed in [26].

2.2 Location-Allocation Problems

This class of problems was formulated by Cooper [12]. They are concerned with supplying a known fixed set of destinations with some commodity, material or service. The requirements at each destination and the shipping costs from any point to another are known. A given number of sources have to be located so as to satisfy the requirements at the destinations in an optimal manner. There are capacity restrictions on each source. Costs are proportional to distances. Typical examples are location of warehouses, distribution centers, communication centers, and machines or other production facilities. In the case where distances are rectilinear, the problem can be formulated as BLP 1. We will now show how this can be done.

Suppose the given coordinates of the destinations are (d_j, e_j), $j = 1,\ldots,n$. Let r_j, $j = 1,\ldots,n$, represent the requirements at the destinations and c_i, $i = 1,\ldots,m$, be the given capacities of the m sources. We are required to find the coordinates (x_i, y_i) of the sources and the quantity w_{ij} that is to be shipped from source i to destination j so as to minimize costs. The problem can be stated mathematically as:

$$\begin{aligned} \text{P5:}\quad \text{Minimize} \quad & \sum_{i=1}^{m} \sum_{j=1}^{n} w_{ij}[|d_j - x_i| + |e_j - y_i|] \\ \text{Subject to} \quad & \sum_{j=1}^{n} w_{ij} \le c_i; \quad i = 1,\ldots,m, \\ & \sum_{i=1}^{m} w_{ij} = r_j; \quad j = 1,\ldots,n, \\ & w_{ij} \ge 0 \end{aligned}$$

Let

$$\begin{aligned} d_j - x_i &= x_i^+ \quad \text{if } x_i \le d_j \\ x_i - d_j &= x_i^- \quad \text{if } x_i \ge d_j \\ e_j - y_i &= y_i^+ \quad \text{if } y_i \le e_j \\ y_i - e_j &= y_i^- \quad \text{if } y_i \ge e_j. \end{aligned}$$

The problem P5 now reduces to

$$\text{P6:}\quad \text{Minimize} \quad \sum_{i=1}^{m}\sum_{j=1}^{n} w_{ij}[x_i^+ + x_i^- + y_i^+ + y_i^-]$$

$$\text{Subject to} \quad x_i^+ - x_i^- = d_j - x_i$$

$$y_i^+ - y_i^- = e_j - y_i$$

$$\text{for } i = 1,\ldots,m \text{ and } j = 1,\ldots,n.$$

$$\sum_{j=1}^{n} w_{ij} \le c_i, \quad i = 1,\ldots,m$$

$$\sum_{i=1}^{m} w_{ij} = r_j, \quad j = 1,\ldots,n$$

$$x_i^+,\ x_i^-,\ y_i^+,\ y_i^-,\ x_i,\ y_i,\ w_{ij} \ge 0$$

where the constraints in x and y define X_0 and the constraints in w_{ij} define Y_0. In problem P6, we need not write the conditions $(x_i^+ \cdot x_i^-) = 0$ and $(y_i^+ \cdot y_i^-) = 0$. These conditions will be satisfied at optimality since $w_{ij} \ge 0$ [41]. By suitably choosing the origin of the coordinate system, we can impose the conditions $x_i \ge 0$, $y_i \ge 0$.

Problem P6 is of the form BLP 1. The Transportation-Location problem is a nonconvex programming problem. It is a generalization of some simple models. If the location of the sources is fixed, the problem reduces to a transportation problem, which has been well solved, see, for example, [27]. If the quantities to be sent from each source to each destination are fixed, the problem reduces to an absolute value problem, which can be solved by linear programming [47]. If both the location of the sources and the allocation of amounts to be shipped are variable, the problem becomes extremely difficult to solve. Heuristic methods have been proposed by Cooper in [14] and enumeration of all extreme points of the transportation problem in [13], since it is shown that the optimum will be attained at an extreme point. Transportation-location problems involving Euclidean distances are also discussed in [13].

We observe that the set Y_0 of BLP 1 corresponds to a transportation problem in a Transportation-Location problem. As mentioned in Chapter I, one of the methods used to solve nonconvex problems is by cutting planes. However, this will tend to destroy both the separable structure of the problem and the transportation format unless the cutting planes are introduced in a special way. In the cutting plane algorithm that we will be developing, we will maintain both these properties.

The X_0-set of the Transportation-Location problem has a block diagonal structure with coupling variables. In our cutting plane algorithm, we will be adding cutting planes to the X_0-set. This will lead to a block diagonal structure with both coupling constraints and variables. Decomposition techniques such as Ritter's [28] partitioning procedure can be used to solve a linear programming problem over X_0 which the cutting plane algorithm requires. This can provide the capability of solving large problems.

We will also be developing a second algorithm for BLP 1 in which no cutting planes are added. It involves the construction of a sequence of polytopes. In this algorithm, it will be necessary to solve linear programs over X_0. We can now use Rosen's partitioning procedure [28] to decompose the problem.

For both the algorithms, it will be necessary to repeatedly solve a parametric linear program in Y_0. For the Transportation-Location problem, this reduces to a parametric transportation problem. Srinivasan and Thompson [43] have shown how this can be done very efficiently. We thus expect that the algorithms that we will be developing will have the capability of solving realistic Transportation-Location problems.

2.3 Maximization of a Convex Quadratic Function

In production planning problems, it may be more realistic to assume that profit is a convex function of the level of production. This may be because of economies of scale. As the level of production is increased, it is expected that profits will increase more rapidly than in direct proportion because the fixed costs remain virtually constant. Soland [42] discusses examples of facility location and transportation problems in which cost functions are concave. We will now show how a convex quadratic maximum problem can be transformed into a Bilinear Problem.

Let us consider the problem:

$$\begin{aligned} \text{P7:}\quad & \text{Maximize} && f(z) = 2\,c^t z + z^t Q z \\ & \text{Subject to} && Az \le b \\ & && z \ge 0 \end{aligned}$$

where Q is a positive semi-definite matrix and the constraint set is nonempty and bounded.

Theorem 2.1. Let $(\bar{x},\bar{y})$ be an optimal solution of the problem

$$\begin{aligned} \text{P8:} \quad \text{Maximize} \quad & \phi(x,y) = c^t x + c^t y + x^t Qy \\ \text{Subject to} \quad & Ax \le b, \quad x \ge 0 \\ & Ay \le b, \quad y \ge 0. \end{aligned}$$

Then both $\bar{x}$ and $\bar{y}$ solve P7. If Q is positive definite, then $\bar{x} = \bar{y}$.

Proof. See [1].

This problem has twice as many constraints as the original problem. Nevertheless, the cutting plane algorithm that we have developed proved more efficient than Moreno's [33] algorithm for maximizing a convex function. Moreover, Konno [26] has shown that only one simplex tableau need ever be stored. Hence the increase in the number of constraints is not a disadvantage.

2.4 Orthogonal Production Scheduling

Let us consider a multi-stage production system:

$$\begin{aligned} \text{P9:} \quad \text{Minimize} \quad & \sum_{\ell=1}^{L} \sum_{j=1}^{n} c_j^\ell x_j^\ell \\ \text{Subject to} \quad & \sum_{j=1}^{n} a_{ij} x_j^\ell \ge b_i^\ell, \quad i = 1,\dots,m;\ \ell = 1,\dots,L. \\ & x_j^\ell \ge 0, \quad j = 1,\dots,n;\ \ell = 1,\dots,L. \end{aligned}$$

where b_i^ℓ = demand for commodity i at period ℓ.

c_j^ℓ = unit cost associated with activity j at period ℓ.

a_{ij} = technological coefficients.

x_j^ℓ = level of activity j at period ℓ.

In addition, suppose we have the restriction that certain activities cannot be used in two consecutive periods because of, for example, maintenance of machines. Thus we require that $(x_j^{\ell-1}.x_j^\ell) = 0$, $\ell = 2,\ldots,L$.

To write this as a bilinear problem, we can use the penalty function method. Let M be a large positive constant and let x^i represent the vector of variables for period i. Then we can write P9 and the orthogonal restriction $(x^{i-1})^t x^i = 0$ as:

$$
\begin{aligned}
\text{Minimize} \quad & \sum_{\ell=1}^{L} (c^\ell)^t x^\ell + M \sum_{\ell=2}^{L} (x^{\ell-1})^t x^\ell \\
\text{Subject to} \quad & Ax^\ell \geq b^\ell, \quad \ell = 1,\ldots,L \\
& x^\ell \geq 0.
\end{aligned}
$$

For L = 2, this problem is of the form BLP 1. It is not very clear how a cutting plane algorithm will compare with, for example, a branch and bound procedure. The large penalty M has the effect of generating very shallow cuts.

2.5 Application to Game Theory

Suppose there are two players, P_1 and P_2. Suppose P_1 selects his strategy first. He can select a vector x from the constraint set $X = \{x|A_1x\leq b,\ x\geq 0\}$. Depending on the strategy x selected by P_1, P_2 can select his strategy y from the set $Y(x) = \{y|A_2y\leq d+Cx,\ y\geq 0\}$. Y(x) is assumed nonempty and bounded for all $x \in X$. When P_1 chooses the strategy x and P_2 chooses y, there is a payoff from P_1 to P_2 given by $f(x,y) = p^t x + q^t y$, where p and q are given vectors. P_2 will naturally try to maximize $f(x,y)$ over $y \in Y(x)$ and hence will solve the problem

$$\begin{array}{ll} \text{Maximize} & q^t y \\ \text{Subject to} & y \in Y(x). \end{array}$$

On the other hand, player P_1 will choose $x \in X$ so as to minimize $f(x,y(x)) = p^t x + q^t y(x)$. Thus he will solve the problem

$$\begin{array}{ll} \text{Minimize} & p^t x + q^t y(x) \\ \text{Subject to} & x \in X. \end{array}$$

Now

$$\begin{aligned} q^t y(x) &\equiv \max_y \{q^t y \mid A_2 y \le d + c^t x,\ y \ge 0\} \\ &= \min_z \{(d+Cx)^t z \mid A_2 z \ge q,\ z \ge 0\} \end{aligned}$$

Hence the problem which P_1 will solve becomes:

$$\begin{array}{ll} \text{Minimize} & \left(p^t x + \min\{(d+Cx)^t z \mid A_2^t z \ge q,\ z \ge 0\}\right) \\ \text{Subject to} & A_1 x \le b, \quad x \ge 0 \end{array}$$

which is equivalent to the problem

$$\text{Minimize} \quad \phi(x,z) = d^t z + p^t x + z^t Cx$$

$$\text{Subject to} \quad A_2 z \geq Q, \quad z \geq 0$$

$$A_1 x \leq b, \quad x \geq 0$$

which is a bilinear programming problem.

2.6 Multi-Stage Assignment Problem

We will first give an example of a two-stage Assignment Problem. Suppose we have N jobs and N machines. Further, suppose the profit associated with an assignment of machine i to job j at the second stage depends on whether or not machine i was assigned to some job k at the previous stage. If it was indeed assigned, then the profit at the second stage is $\bar{p}_{ij} + q_{ijk}$. The profit at the first stage when machine i is assigned to job k is simply p_{ik}. Thus the total two-stage profit is given by

$$\sum_{i=1}^{N} \sum_{j=1}^{N} p_{ij} x_{ij}^{1} + \sum_{i=1}^{N} \sum_{j=1}^{N} p_{ij} x_{ij}^{2} + \sum_{i=1}^{N} \sum_{j=1}^{N} \sum_{k=1}^{N} q_{ijk} x_{ij}^{2} x_{ik}^{1}.$$

The constraints represent the restrictions that one and only one machine can be assigned to each job. Thus we have the following constraints for the two stages:

$$\sum_{j=1}^{N} x_{ij}^{p} = 1, \quad i = 1,\ldots,N.$$

$$\sum_{i=1}^{N} x_{ij}^{p} = 1, \quad j = 1,\ldots,N.$$

$$x_{ij}^p = 0,1 \quad \text{for all } (i,j)$$

$$p = 1,2.$$

Let

$$X_0^p = \{(x_{11}^p,\ldots,x_{NN}^p) \mid \sum_{j=1}^{N} x_{ij}^p=1,\ i=1,\ldots,N,$$
$$\sum_{i=1}^{N} x_{ij}^p=1,\ j=1,\ldots,N,$$
$$x_{ij}^p \geq 0\}$$

for p = 1,2. Ignoring the 0,1 restriction on the variables, we can write the two-stage assignment problem as:

$$\begin{array}{ll} \text{Maximize} & p^t x^1 + \bar{p}^t x^2 + (x^1)^t Q x^2 \\ \text{Subject to} & x^1 \in X_0^1 \\ & x^2 \in X_0^2 \end{array}$$

The problem as stated above without the 0,1 restriction is a Bilinear Programming problem. We will show in Chapter III that an optimal solution to a Bilinear Problem is attained at an extreme point of X_0^1 and of X_0^2. At each extreme point of an assignment problem all the variables are 0 or 1. Thus we need not specify the 0-1 requirement since it will automatically hold at an optimal solution to the Bilinear Problem.

We have shown how a two-stage assignment problem can be formulated as a Bilinear Problem. One can show that a multi-stage assignment problem can also be formulated as a Bilinear Problem.

2.7 Application to Decision Theory

In Section 5.8, we have described a problem in Decision Theory. Given a decision tree, each path through the tree represents a strategy with a utility value associated with it. The objective is to maximize the expected utility over a finite set of vectors, each representing the values associated with a strategy. The problem reduces to one of maximizing a linear function over a polytope, where the polytope is defined as the convex hull of a finite set of points. In Chapter V, we will develop an algorithm for generating a sequence of polytopes. Each polytope is defined as the convex hull of a finite set of points. This algorithm can be used to solve the problem of maximizing the expected utility.

2.8 Other Applications

It is shown in [37] that a 0-1 programming problem can be transformed to a concave programming problem by the penalty function method and then solved by a cutting plane algorithm. Using Theorem 2.1, we can transform the concave problem to a Bilinear Problem. Since a fixed charge problem can be written as a 0-1 programming problem, we can transform a fixed charge problem to a Bilinear Problem. However, implicit enumeration schemes for 0-1 programs have proven to be much more efficient than cutting plane approaches.

CHAPTER III

PRELIMINARY PROPERTIES AND DETERMINATION OF A PSEUDO-GLOBAL MINIMUM

3.1 Introduction

In this chapter, we will be discussing the properties of the Bilinear Programming Problem and the nature of its objective function. This will explain the difficulties involved in solving this problem, and highlight the special properties that the problem possesses which we would like to exploit in obtaining a global minimum to the problem. For the two algorithms that we will develop later on, it is necessary to first find a feasible solution to the problem that has certain special properties. This point is the pseudo-global minimum and we will show how such a point can be obtained. Finally, we will show how a good feasible solution can be determined. This may provide faster convergence of the algorithms, and will be useful in solving large problems.

3.2 Preliminary Properties

The Bilinear Problem BLP 1 was stated in Section 1.2. It can also be expressed as a quadratic programming problem:

$$\text{(BLP 2):} \quad \text{Minimize} \quad f(z) = q^t z + \tfrac{1}{2} z^t Q z,$$

$$\text{Subject to} \quad Az \le b, \quad z \ge 0$$

where $z^t = (x^t, y^t)$, $q^t = (c^t, d^t)$, and

$$Q = \begin{bmatrix} 0 & C \\ C & 0 \end{bmatrix}, \quad A = \begin{bmatrix} E & 0 \\ 0 & F \end{bmatrix}, \quad b = \begin{bmatrix} e \\ f \end{bmatrix}$$

However, we will work with problem BLP 1 so as to take advantage of its special structure and the resulting properties.

Since the constraints of BLP 1 are linear, it would seem natural to try to explore whether or not adjacent extreme point methods could be suitably modified to solve the bilinear problem. In order for this approach to succeed, the problem must have the following two characteristics:

1. The optimum is attained at an extreme point.
2. A local minimum is a global minimum.

It turns out that the bilinear problem has the first property but not the second. We first determine the nature of the objective function $\phi(x,y)$ of BLP 1.

That ϕ is nonconcave is demonstrated by the following example. Let $x \in R^1$, $y \in R^1$, $c = 0$, $d = 0$, $C = I$, $x_1 = 2$, $y_1 = 2$, $x_2 = 1$, $y_2 = 1$, $\lambda = 1/2$. Then $\lambda x_1 y_1 + (1-\lambda)x_2 y_2 = (1/2)4 + (1/2)1 = 5/2 \nleq [\lambda x_1 + (1-\lambda)x_2][\lambda y_1 + (1-\lambda)y_2] = [(1/2)2+(1/2)1][(1/2)2+(1/2)1] = 9/4$. Similarly, by picking the points $x_1 = 1$, $y_1 = 2$, $x_2 = 2$, $y_2 = 1$, one can show that ϕ is nonconvex. It is well known that convexity and concavity requirements on ϕ can be relaxed to a certain degree and ϕ will still attain a minimum at an extreme point of the polytope on which it is defined, and every local minimum of ϕ will be a global minimum. We now investigate these conditions.

Definition 3.1. A function f defined over a convex set S is *quasiconcave* on S if for all x^1 and $x^2 \in S$, $f[\lambda x^1+(1-\lambda)x^2] \geq \min[f(x^1),f(x^2)]$, $0 \leq \lambda \leq 1$. The negative of a quasiconcave function is quasiconvex.

Definition 3.2. A quasiconvex function f defined over a convex set S is explicitly quasiconvex on S if for all x^1 and $x^2 \in S$ with $f(x^1) \neq f(x^2)$, $f[\lambda x^1+(1-\lambda)x^2] < \max[f(x^1),f(x^2)]$, $0 < \lambda < 1$.

The following two theorems define the relationship between the nature of a function and the properties of extreme point optimality and the existence of local minima different from the global minimum.

Theorem 3.3. A continuous function f defined over a polytope L attains its minimum at an extreme point of L and of all its convex polyhedral subsets if and only if it is quasiconcave on L.

Proof. See [31].

Theorem 3.4. A continuous function f defined over a polytope L is such that each local minimum is also a global minimum on L and on all its convex polyhedral subsets if and only if f is explicitly quasiconvex on L.

Proof. See [31].

In order to check whether or not the objective function ϕ of BLP 1 or, equivalently, the function f of BLP 2 is quasiconcave and explicitly quasiconvex, we need the following definitions and theorems.

Definition 3.5. A matrix D is *positive semidefinite* if $x^t Dx \geq 0$ for all x.

Theorem 3.6. The quadratic form $\phi(x) = x^t Dx$ is convex on E^n if and only if D is positive semidefinite.

Proof. See [32].

Definition 3.7. A matrix D is *positive subdefinite* if $x^t Dx < 0$ implies $Dx \geq 0$ or $Dx \leq 0$. A quadratic form $\phi(x) = x^t Dx$ is said to be *positive subdefinite* if D is positive subdefinite.

Since any positive semidefinite matrix satisfies the above implication by default, and we would like to exclude positive semidefinite matrices from the class of subdefinite matrices, the following definition is given.

Definition 3.8. A matrix D is *merely* positive subdefinite if it is positive subdefinite but not positive semidefinite. A quadratic form $\phi(x) = x^t Dx$ is merely positive subdefinite if D is merely positive subdefinite.

Theorem 3.9. The quadratic form $\phi(x) = x^t Dx$ is quasiconvex on the nonnegative orthant, E^n_+, if and only if it is positive subdefinite.

Proof. See [32].

The following two theorems provide a computational procedure for checking whether or not a matrix is positive subdefinite and, consequently, to verify whether the associated quadratic form is quasiconvex on the nonnegative orthant or not.

Theorem 3.10. The quadratic form $\phi(x) = x^t Dx$ is merely positive subdefinite if and only if

i. $D \leq 0$ and $D \neq 0$.

ii. D has nonpositive principal minors.

Proof. See [15].

Since the sum of a quasiconvex function and a convex function need not be quasiconvex, and our objective is to determine the quasiconvexity of the quadratic function f of BLP 2, we need an extension of Theorem 3.9 for the general case. This is provided by the following theorem.

Theorem 3.11. If the quadratic function $\phi(x) = c^t x + \frac{1}{2} x^t Dx$ is not convex on E^n, then ϕ is quasiconvex on E^n_+ if and only if the quadratic form

$$\psi(x,\xi) = \frac{1}{2}\begin{pmatrix} x \\ \xi \end{pmatrix}^t \begin{pmatrix} D & c \\ c^t & 0 \end{pmatrix} \begin{pmatrix} x \\ \xi \end{pmatrix}$$

is quasiconvex on E^{n+1}_+.

Proof. See [15].

We have verified that the function f of BLP 2 is not concave on E^n. To determine whether or not f is quasiconvex on E^n_+, we need to check merely positive subdefiniteness of the matrix

$$\begin{bmatrix} Q & q \\ q^t & 0 \end{bmatrix} = \begin{bmatrix} 0 & C & c \\ C & 0 & d \\ c^t & d^t & 0 \end{bmatrix}$$

Since no assumptions have been made on the signs of the matrix C or the vectors c and d in BLP 1, the matrix D in general will not satisfy condition (i) of Theorem 3.10. Thus from Definition 3.8 and Theorem 3.10, D is not merely positive subdefinite. Also, since f is not convex, from Theorem 3.6, D is not positive semidefinite. Hence from Definition 3.8, D is not positive subdefinite. Thus from Theorem 3.9, the function f, and hence the function ϕ of BLP 1 need not be quasiconvex or quasiconcave. Since an explicitly quasiconvex function is quasiconvex, we conclude that ϕ is not explicitly quasiconvex.

Theorems 3.3 and 3.4 need to be explained further. A continuous function f quasiconcave on a polytope L will attain its minimum at a vertex of L. But a continuous function f which attains its minimum at an extreme point of L need not be quasiconcave. An arbitrary function may attain its minimum at an extreme point of L because of the special structure of L. The requirement for quasiconcavity is the existence of the extreme point property for all possible polyhedral subsets of L. Similarly, while explicit quasiconvexity guarantees the fact that there will not exist local minima different from the global minimum, the reverse implication will be true only if this property holds for all polyhedral subsets of L. Hence, we have not eliminated the possibility of the function ϕ of BLP 1 attaining its minimum at an extreme point of its constraint set because of some special properties of the set. Similarly, it is still possible that every local minimum of ϕ is a global minimum of ϕ on the defining polyhedral set. We next show that the first statement is, in fact, a true one, but not the second.

Theorem 3.12. The Bilinear problem BLP 1 has an optimal solution $(\bar{x},\bar{y})$ such that $\bar{x}$ and $\bar{y}$ are extreme points of X_0 and Y_0, respectively.

Proof. For an arbitrary y, consider the program

$$\underset{x\epsilon X_0}{\text{Minimize}} \quad \phi(x,y) = \underset{x\epsilon X_0}{\text{Minimize}} \quad c^t x + d^t y + x^t C y$$

Since this is a linear program, it has an extreme point solution $\bar{x}$ with $\phi(\bar{x},y) \leq \phi(x,y)$ for all $x \in X_0$. Now consider the program

$$\underset{y\epsilon Y_0}{\text{Minimize}} \quad \phi(\bar{x},y) = \underset{y\epsilon Y_0}{\text{Minimize}} \quad c^t \bar{x} + d^t y + \bar{x}^t C y$$

Again this is a linear program with an optimal extreme point solution $\bar{y}$ with $\phi(\bar{x},\bar{y}) \leq \phi(\bar{x},y)$ for all $y \in Y_0$. But we have shown that $\phi(\bar{x},y) \leq \phi(x,y)$ for all $x \in X_0$. Hence $\phi(\bar{x},\bar{y}) \leq \phi(x,y)$ for all $x \in X_0$ and $y \in Y_0$, that is, $(\bar{x},\bar{y})$ is an optimal solution with $\bar{x}$ an extreme point of X_0 and $\bar{y}$ an extreme point of Y_0.

It can be seen that the above theorem need not be true if the problem BLP 1 has common constraints involving both x and y variables. Thus ϕ will not attain its minimum at an extreme point of all possible polyhedral subsets of the polytope on which it is defined.

We now come to the question of local and global minimum.

Definition 3.13. A function f defined over a set X has a *local minimum* at a point $\bar{x}$ if $f(\bar{x}) \le f(x)$ for all $x \in B_\varepsilon(\bar{x}) \cap X$, where $B_\varepsilon(\bar{x})$ is an ε-neighborhood around $\bar{x}$.

Definition 3.14. A function f defined over a set X has a *global minimum* at a point $\bar{x}$ if $f(\bar{x}) \le f(x)$ for all $x \in X$.

Definition 3.15. An *edge* of a convex polyhedron is the line segment joining two extreme points such that no point on the segment can be expressed as a convex combination of two other points in the polyhedron but not on the segment. In this case the two extreme points are said to be *adjacent* to each other.

Given an extreme point $\bar{x}$ of a convex polyhedron, we can write down a basic feasible solution corresponding to it from the system of equations defining the polyhedron. If $\bar{x}$ is a nondegenerate vertex, we can determine any one of its adjacent extreme points by a single pivot operation so as to make basic a variable that is currently nonbasic. In the degenerate case, more than one pivot operation may be required to generate some adjacent extreme points of $\bar{x}$. We will represent the set of all adjacent extreme points of $\bar{x}$ in X_0 by $N(\bar{x})$.

Definition 3.16. Let f be a function defined over a convex polyhedral set X. Then an extreme point $\bar{x}$ of X is called a *local star minimum* if $f(\bar{x}) \le f(x)$ for each $x \in N(\bar{x})$.

Since we know the minimum of BLP 1 is attained at an extreme point, it would seem logical to apply the iterative procedure suggested in the proof of Theorem 3.12. Unfortunately, such a procedure can converge to a local star minimum which is different from the global minimum. Before we give an example to show this, we need the following theorem.

Theorem 3.17. $(\bar{x},\bar{y})$ is an extreme point of $Z = \{(x,y) | x \in X_0, y \in Y_0\}$ if and only if $\bar{x}$ is an extreme point of X_0 and $\bar{y}$ is an extreme point of Y_0.

Proof. Let $(\bar{x},\bar{y})$ be an extreme point of Z. Suppose $\bar{x}$ is not an extreme point of X_0. Then $\bar{x} = \sum_{i=1}^{p} \lambda_i x^i$, $0 < \lambda_i < 1$, $\sum_{i=1}^{p} \lambda_i = 1$, where $x^1,\ldots,x^p$ are extreme points of X_0. Then $(\bar{x},\bar{y}) = \sum_{i=1}^{p} \lambda_i (x^i,\bar{y})$, $0 < \lambda_i < 1$, $\sum_{i=1}^{p} \lambda_i = 1$. But $(x^i,\bar{y}) \in Z$, $i = 1,\ldots,p$. Thus we have expressed $(\bar{x},\bar{y})$ as a convex combination of other points $(x^i,\bar{y}) \in Z$, which contradicts the assumption that $(\bar{x},\bar{y})$ is an extreme point of Z. Hence $\bar{x}$ is an extreme point of X_0. Likewise, $\bar{y}$ is an extreme point of Y_0.

To prove the converse, let $\bar{x}$ and $\bar{y}$ be extreme points of X_0 and Y_0, respectively. Suppose $(\bar{x},\bar{y})$ is not an extreme point of Z. Then $(\bar{x},\bar{y}) = \sum_{i=1}^{r} \gamma_i (x^i,y^i) = \left(\sum_{i=1}^{r} \gamma_i x^i, \sum_{i=1}^{r} \gamma_i y^i\right)$, where (x^i,y^i) are extreme points of Z, $0 < \gamma_i < 1$, $\sum_{i=1}^{r} \gamma_i = 1$. This implies that $\bar{x} = \sum_{i=1}^{r} \gamma_i x^i$. Since $\bar{x}$ is an extreme point, we must have some $\gamma_i = 1$, a contradiction since $0 < \gamma_i < 1$. Hence $(\bar{x},\bar{y})$ is an extreme point of Z.

Corollary 3.18. Each adjacent extreme point of $(\bar{x},\bar{y}) \in Z$ is either of the form $(\bar{x},y^i)$, $y^i \in N(\bar{y})$ or of the form $(x^i,\bar{y})$, $x^i \in N(\bar{x})$.

Proof. This follows from Theorem 3.17 and the discussion following Definition 3.15 of an adjacent extreme point.

Suppose we now develop the following iterative procedure, a precise statement of which will be made later. Choosing an arbitrary point $x^1 \in X_0$, we solve the linear programming problem: Min $c^t x^1 + d^t y + x^{1t} Cy$, $y \in Y_0$. Let the extreme point optimal solution be attained at y^1. Next we solve the linear programming problem: Min $c^t x + d^t y^1 + x^t Cy^1$, $x \in X_0$. Suppose the optimal solution is at x^2. Once again a linear problem in y is solved and so on till we obtain a pair of points $(\bar{x},\bar{y})$ such that $\bar{x}$ is the solution with y fixed at $\bar{y}$ and $\bar{y}$ is the solution with x fixed at $\bar{x}$. From Corollary 3.18, an adjacent extreme point to $(\bar{x},\bar{y})$ in Z will be either $(\bar{x},y^i)$, $y^i \in N(\bar{y})$ or $(x^i,\bar{y})$, $x^i \in N(\bar{x})$. We now show that $(\bar{x},\bar{y})$ is a local star minimum.

Lemma 3.19. $(\bar{x},\bar{y})$ is a local star minimum of BLP 1 if and only if $\bar{x}$ is a solution to

$$\text{P9:} \quad \text{Minimize} \quad \phi(x,\bar{y}) = c^t x + d^t \bar{y} + x^t C\bar{y}$$
$$\text{Subject to} \quad x \in X_0$$

and $\bar{y}$ is a solution to

$$\text{P10:} \quad \text{Minimize} \quad \phi(\bar{x},y) = c^t\bar{x} + d^ty + \bar{x}^tCy$$
$$\text{Subject to} \qquad y \in Y_0.$$

Proof. Let $\bar{x}$ solve P9 and $\bar{y}$ solve P10. Then we have $\phi(\bar{x},\bar{y}) \le \phi(x,\bar{y})$ for each $x \in X_0$. In particular, $\phi(\bar{x},\bar{y}) \le \phi(x^i,\bar{y})$ for each $x^i \in N(\bar{x})$. Similarly, $\phi(\bar{x},\bar{y}) \le \phi(\bar{x},y^i)$ for each $y^i \in N(\bar{y})$. From Corollary 3.18, each adjacent extreme point of $(\bar{x},\bar{y})$ in $Z = \{(x,y)|x \in X_0, y \in Y_0\}$ is of the form $(\bar{x},y^i)$, $y^i \in N(\bar{y})$ or $(x^i,\bar{y})$, $x^i \in N(\bar{x})$. Thus $\phi(\bar{x},\bar{y}) \le \phi(x,y)$, for each $(x,y) \in N(\bar{x},\bar{y})$. Hence $(\bar{x},\bar{y})$ is a local star minimum.

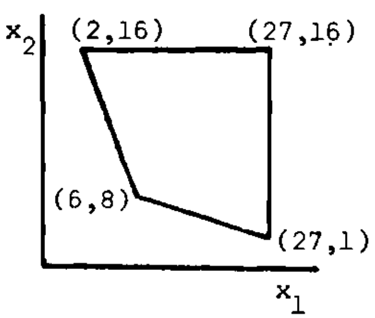
Conversely, let $(\bar{x},\bar{y})$ be a local star minimum. Hence we have $\phi(\bar{x},\bar{y}) \le \phi(\bar{x},y^i)$ for each $y^i \in N(\bar{y})$. Let us consider problem P10. Suppose we apply the simplex method to P10 and have a basic feasible solution corresponding to the point $\bar{y}$. Since $\phi(\bar{x},\bar{y}) \le \phi(\bar{x},y^i)$ for each $y^i \in N(\bar{y})$, the simplex method will terminate and $\bar{y}$ is a solution to P10. Similarly, by considering problem P9, we can show that if $(\bar{x},\bar{y})$ is a local star minimum, then $\bar{x}$ solves P9.

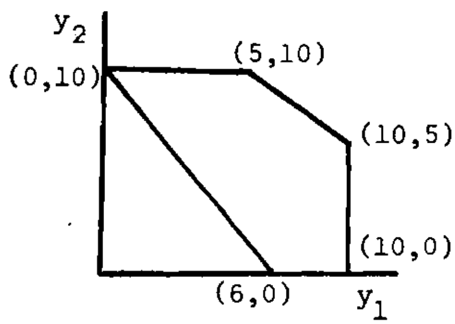
We will provide other equivalent characterizations of a local star minimum later in Theorem 3.27 and Corollary 3.28. We will now show by an example that a local star minimum need not be a global minimum. We will be using this example to illustrate other features later on.

Example Problem 1

$$\begin{array}{llr}
\text{Min} & x_1y_1 + x_2y_2 & \\
\text{Subject to} & x_1 + 3x_2 & \geq 30 \\
& 2x_1 + x_2 & \geq 20 \\
& x_1 & \leq 27 \\
& x_2 & \leq 16 \\
& \frac{10}{6} y_1 + y_2 & \geq 10 \\
& y_1 + y_2 & \leq 15 \\
& y_1 & \leq 10 \\
& y_2 & \leq 10
\end{array}$$

$$x_1,\ y_1,\ x_2,\ y_2 \geq 0.$$





Value of Objective Function Z at Extreme Points

x_1	x_2	y_1	y_2	Z	x_1	x_2	y_1	y_2	Z
2	16	0	10	160	27	1	0	10	10
		5	10	170			5	10	145
		10	5	100			10	5	275
		10	0	20			10	0	270
		6	0	12			6	0	162
27	16	0	10	160	6	8	0	10	80
		5	10	295			5	10	110
		10	5	350			10	5	100
		10	0	270			10	0	60
		6	0	162			6	0	36

Let $\bar{x} = (2,16)^t$, $\bar{y} = (6,0)^t$, $x^* = (27,1)^t$, $y^* = (0,10)^t$. Then $(\bar{x},\bar{y})$ is a local star minimum whereas (x^*,y^*) is the global minimum to the above problem.

The existence of local star minima different from the global minimum is the essential difficulty in the problem, and the reason why adjacent extreme point methods do not work. It is intuitively clear, however, that these methods will play an important role in locating the global minimum.

To characterize certain extreme points of BLP 1, we need to transform the origin of the coordinate system to another extreme point of the feasible set, and we now show how this can be done.

3.3 Transfer of Origin and Resolution of Degeneracy

To characterize certain extreme points of BLP 1, we need to transform the origin of the coordinate system to another extreme point of the feasible set. Further, given an extreme point solution with m basic variables, in the absence of degeneracy, there will be precisely m edges incident on the extreme point. If the solution were degenerate, there could conceivably be more than m edges. As we shall see later on, it is important that we identify precisely m edges incident on the extreme point under consideration. These can then correspond with the coordinate axes at this point. For this purpose, we will use the procedure developed in [2].

Let $(\bar{x},\bar{u})$ be a basic solution to the system

$$Ex + u = e \qquad x \geq 0,\ u \geq 0$$

which define the set X_0. By suitably rearranging the basic and non-basic variables at $(\bar{x},\bar{u})$, we can write this as

$$\begin{bmatrix} E_{11} & E_{12} \\ E_{21} & E_{22} \end{bmatrix} \begin{bmatrix} x_B \\ x_N \end{bmatrix} + \begin{bmatrix} I & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} u_N \\ u_B \end{bmatrix} = \begin{bmatrix} e^1 \\ e^2 \end{bmatrix} \tag{1}$$

where (x_B,u_B) are the basic variables. This can be rewritten as

$$\begin{bmatrix} E_{11} & 0 \\ E_{21} & I \end{bmatrix} \begin{bmatrix} x_B \\ u_B \end{bmatrix} + \begin{bmatrix} E_{12} & I \\ E_{22} & 0 \end{bmatrix} \begin{bmatrix} x_N \\ u_N \end{bmatrix} = \begin{bmatrix} e^1 \\ e^2 \end{bmatrix} \tag{2}$$

Multiplying by the basis inverse, namely

$$\begin{bmatrix} E_{11}^{-1} & 0 \\ -E_{21}E_{11}^{-1} & I \end{bmatrix}$$

we get

$$\begin{bmatrix} x_B \\ u_B \end{bmatrix} + \begin{bmatrix} E_{11}^{-1}E_{12} & E_{11}^{-1} \\ E_{22}-E_{21}E_{11}^{-1}E_{12} & -E_{21}E_{11}^{-1} \end{bmatrix} \begin{bmatrix} x_N \\ u_N \end{bmatrix} = \begin{bmatrix} E_{11}^{-1}e^1 \\ e^2-E_{21}E_{11}^{-1}e^1 \end{bmatrix} \tag{3}$$

We will first show how to identify precisely m edges incident on a degenerate vertex $\bar{x}$ of X_0. Later on, we will examine the form of the Bilinear Problem resulting from a transformation of the origin to any given vertex $(\bar{x},\bar{y})$, $\bar{x} \in X_0$, $\bar{y} \in Y_0$.

From Equation (3) and writing $x_N = x_N$, we get

$$x_B = E_{11}^{-1} e^1 - E_{11}^{-1}E_{12}x_N - E_{11}^{-1}u_N \qquad (4)$$

$$x_N = \quad 0 \quad - (-I)x_N \quad - 0u_N \qquad (5)$$

Let $J_1 = \{x_i | x_i$ is nonbasic at $(\bar{x},\bar{u})\}$, $J_2 = \{u_i | u_i$ is nonbasic at $(\bar{x},\bar{u})\}$ and $J = J_1 \cup J_2$. Let $P = \{(x^t,u^t)^t | Ex+u=e,\ x\geq 0, u\geq 0\}$. $P \subset R^{m\times k_1}$. Let $p^t = (x^t,u^t)$. Then (4) and (5) can be written as:

$$x = \bar{x} + \sum_{j\in J_1} \bar{e}^j(-x_j) + \sum_{k\in J_2} \bar{e}^k(-u_k) = \bar{x} + \sum_{j\in J} \bar{e}^j(-p_j) \qquad (6)$$

Clearly, a x given by (6) satisfies the inequality $Ex \leq e$. However, it may not satisfy the nonnegativity restriction. By considering the structure of the columns corresponding to $\bar{e}^j$, $j \in J$, in the original tableau, one can see that the columns $\bar{e}_j$ are linearly independent.

In the problem defined by Equation (1) there are $m + k_1$ variables. Equation (1) has k_1 constraints. Hence k_1 variables are basic and m variables are nonbasic. Hence the cardinality of the set J is m. We will be interested in the m halflines (rays) defined by

$$\xi^j = \{x | x=\bar{x}-\bar{e}^j\lambda_j,\ \lambda_j\geq 0\},\ j \in J \qquad (7)$$

Let us consider the polyhedron X_0' derived from X_0 by deleting each constraint of X_0 for which the associated variable u_i or x_j is basic but zero in the optimal solution $(\bar{x},\bar{u})$. More precisely, let M and N be the index sets associated with the constraints $Ex \le e$ and $x \ge 0$, respectively, i.e., $X_0 = \{x\epsilon R^m | \sum_{j=1}^{m} e_{ij}x_j \le e_i,\ i\epsilon M, x_j \ge 0, j\epsilon N\}$. Given the optimal basic feasible solution $(\bar{x},\bar{u})$, let $M^0 = \{i\epsilon M | u_i$ is basic and $\bar{u}_i = 0\}$, $N^0 = \{j\epsilon N | x_j$ is basic and $\bar{x}_j = 0\}$. Then $X_0' = \{x\epsilon R^m | \sum_{j=1}^{m} e_{ij}x_j \le e_i,\ i\epsilon M-M^0; x_j \ge 0, j\epsilon N-N^0\}$. $X_0 \subset X_0'$ since X_0' was obtained by deleting constraints of X_0.

<u>Theorem 3.20</u>. Let $\bar{x}$ be an extreme point solution to the problem: Min $c^t x$, $x \in X_0$. Let X_0' be defined as above. Then $\bar{x}$ is a vertex of X_0' and $c^t x = c^t \bar{x}$ is a supporting hyperplane for X_0'. X_0' has m distinct edges incident on $\bar{x}$, and each halfline (7) contains exactly one such edge.

Proof. See [2].

The polyhedron X_0' has m distinct edges incident on $\bar{x}$ and each edge is a subset of a halfline (7). Instead of working with X_0, we will be working with the set X_0' so that even in the degenerate case we can uniquely identify m edges incident on the vertex $\bar{x}$. Let us also consider the polyhedral cone C with vertex at $\bar{x}$ and whose m extreme rays are given by (7). By definition, any $x \in C$ is given by $x = \bar{x} + \sum_{j\epsilon J} (-\bar{e}^j)\beta_j$, for some $\beta_j \ge 0$. Since each edge of X_0' incident on $\bar{x}$ is contained in (7), it is clear that any element of X_0' is contained in C. We thus have the following set inclusion relationships: $X_0 \subset X_0' \subset C$.

We will now examine the structure of the Bilinear Problem when the origin of the coordinate system is transferred to any given vertex

$(\bar{x},\bar{y})$, $\bar{x} \in X_0$, $\bar{y} \in Y_0$. From Equation (3), since $x_B \geq 0$, $u_B \geq 0$ we get

$$\begin{bmatrix} E_{11}^{-1}E_{12} \\ E_{22}-E_{21}E_{11}^{-1}E_{12} \end{bmatrix} x_N + \begin{bmatrix} E_{11}^{-1} \\ -E_{21}E_{11}^{-1} \end{bmatrix} u_N \leq \begin{bmatrix} E_{11}^{-1}e^1 \\ e^2-E_{21}E_{11}^{-1}e^1 \end{bmatrix} \tag{8}$$

$$x_N \geq 0, \quad u_N \geq 0. \tag{9}$$

The right-hand side of (8) is nonnegative since it represents the values of the basic variables. Now (8) and (9) are of the same structure and dimension as the original inequalities defining X_0. Given specific values for x_N and u_N satisfying (8) and (9) by substituting these values in (3), we can find the values of x_B and u_B which will obviously be nonnegative. By renaming the variables u_N and x_N, we can, therefore, assume that the origin is a feasible point of X_0. Similarly, by defining v as the vector of slack variables, Y_0 can be written as:

$$\begin{bmatrix} F_{11}^{-1}F_{12} \\ F_{22}-F_{21}F_{11}^{-1}F_{12} \end{bmatrix} y_N + \begin{bmatrix} F_{11}^{-1} \\ -F_{21}F_{11}^{-1} \end{bmatrix} v_N \leq \begin{bmatrix} F_{11}^{-1}f^1 \\ f^2-F_{21}F_{11}^{-1}f^1 \end{bmatrix} \tag{10}$$

$$y_N \geq 0, \quad v_N \geq 0 \tag{11}$$

We would now like to determine the form of the objective function ϕ of BLP 1 resulting from the transformation described above. By a conforming partition of c, d, and C, we can write ϕ at x and y as:

$$\phi(x,y) = c_B^t x_B + c_N^t x_N + d_B^t y_B + d_N^t y_N + \begin{bmatrix} x_B \\ x_N \end{bmatrix}^t \begin{bmatrix} C_{11} C_{12} \\ C_{21} C_{22} \end{bmatrix} \begin{bmatrix} y_B \\ y_N \end{bmatrix} \tag{12}$$

From (12) if we eliminate x_B with the help of (3) and y_B with the help of a corresponding equation in terms of y_N and v_N, we get:

$$\bar{\phi}(x_N, y_N, u_N, v_N) = k + (a^1)^t u_N + (a^2)^t x_N + (b^1)^t v_N + (b^2)^t y_N + \begin{bmatrix} u_N \\ x_N \end{bmatrix}^t \begin{bmatrix} \bar{C}_{11} \bar{C}_{12} \\ \bar{C}_{21} \bar{C}_{22} \end{bmatrix} \begin{bmatrix} v_N \\ y_N \end{bmatrix} \tag{13}$$

where $k = c_B^t E_{11}^{-1} e^1 + d_B^t F_{11}^{-1} f^1 + [E_{11}^{-1} e^1]^t [C_{11}][F_{11}^{-1} f^1]$, a constant which is the objective function value at (x,y),

$$a^1 = -(E_{11}^{-1})^t (c_B + C_{11} F_{11}^{-1} f^1)$$

$$a^2 = c_N - (E_{11}^{-1} E_{12})^t c_B + [C_{21} - (E_{11}^{-1} E_{12})^t C_{11}] F_{11}^{-1} f^1$$

$$b^1 = -(F_{11}^{-1})^t (d_B + C_{11} E_{11}^{-1} e^1)$$

$$b^2 = d_N - (F_{11}^{-1} F_{12})^t d_B + [C_{12} - C_{11} F_{11}^{-1} F_{12}]^t [E_{11}^{-1} e^1]$$

$$\bar{C}_{11} = [E_{11}^{-1}]^t C_{11} F_{11}^{-1}$$

$$\bar{C}_{12} = [E_{11}^{-1}]^t [C_{11} F_{11}^{-1} F_{12} - C_{12}]$$

$$\bar{C}_{21} = -[C_{21} - [E_{11}^{-1} E_{12}]^t C_{11}] F_{11}^{-1}$$

$$\bar{C}_{22} = C_{22} - [E_{11}^{-1} E_{12}]^t C_{12} - C_{21} [F_{11}^{-1} F_{12}] + [E_{11}^{-1} E_{12}]^t C_{11} [F_{11}^{-1} F_{12}].$$

Once again we can see that the form of $\bar{\phi}$ is exactly the same as the form of ϕ of BLP 1. We can thus transform the origin of the system of coordinates of BLP 1 to any of its extreme points and maintain all its essential characteristics. We will omit the constant k from the statement of the function ϕ since it does not affect the optimization problem and write the Bilinear Problem in the following form:

$$\begin{aligned} \text{BLP 3:} \quad \text{Minimize} \quad & \phi(x,y) = c^t x + d^t y + x^t C y \\ \text{Subject to} \quad & x \in X_0 = \{x | Ex \le e,\ x \ge 0\},\ (e \ge 0) \\ & y \in Y_0 = \{y | Fy \le f,\ y \ge 0\},\ (f \ge 0) \end{aligned}$$

Without loss of generality we will assume as before that X_0 and Y_0 are bounded.

We observe that this form is exactly the same as BLP 1 except that the parameters e and f are nonnegative, since the right-hand side of (8) is nonnegative.

3.4 Characterization of Extreme Points

In this research, we will be developing two algorithms for finding the global minimum of BLP 1. An initial step in both procedures will be to locate an extreme point of X_0 that has certain special properties. The following characterization of the extreme points will be useful in that context. Even though some theorems are not explicitly used, they are stated for the sake of completeness.

Theorem 3.21. The origin (0,0) of BLP 3 is a global minimum if for each $x \in X_0$ and $y \in Y_0$, $c^t x \ge 0$, $d^t y \ge 0$ and $x^t C y \ge 0$.

Proof. For each $x \in X_0$, $y \in Y_0$ we have

$$\begin{aligned}\phi(x,y) &= c^t x + d^t y + x^t C y \\ &\geq 0 \quad \text{from hypothesis} \\ &= \phi(0,0)\end{aligned}$$

Hence (0,0) is a global minimum of BLP 3.

Corollary 3.22. The origin (0,0) is a global minimum of BLP 3 if $c \geq 0$, $d \geq 0$ and $C \geq 0$.

Proof. Follows from Theorem 3.21 since $x \geq 0$ and $y \geq 0$ for $x \in X_0$ and $y \in Y_0$.

Theorem 3.23. The origin (0,0) of BLP 3 is a local minimum if and only if for each $x \in X_0$, $y \in Y_0$

i. $c^t x \geq 0$ and $d^t y \geq 0$ and

ii. if $x^t C y < 0$, then either $c^t x > 0$ or $d^t y > 0$.

Proof. Suppose conditions (i) and (ii) hold. Let $\hat{x} \in X_0$, $\hat{y} \in Y_0$ be such that $\hat{x}^t C\hat{y} < 0$, $c^t\hat{x} > 0$, $d^t\hat{y} \geq 0$. Let $\varepsilon_1 = \min\{c^t\hat{x}/-\hat{x}^t C\hat{y}, 1\}$. Clearly $0 < \varepsilon_1 \leq 1$. Hence for $0 < \varepsilon \leq \varepsilon_1$, $(\varepsilon\hat{x}) \in X_0$, $(\varepsilon\hat{y}) \in Y_0$, and

$$\begin{aligned}\phi(\varepsilon\hat{x},\varepsilon\hat{y}) &= \varepsilon c^t\hat{x} + \varepsilon d^t\hat{y} + \varepsilon^2\hat{x}^t C\hat{y} \\ &\geq \varepsilon(c^t\hat{x}+\varepsilon\hat{x}^t C\hat{y}) \\ &\geq 0 \text{ since } 0 < \varepsilon \leq \frac{c^t\hat{x}}{-\hat{x}^t C\hat{y}} \\ &= \phi(0,0)\end{aligned}$$

Similarly, one can show that $x^tCy < 0$, $c^tx \geq 0$, $d^ty > 0$ imply that $\phi(\varepsilon x,\varepsilon y) \geq \phi(0,0)$. Hence $(0,0)$ is a local minimum.

Conversely, let $(0,0)$ be a local minimum. If $c^tx < 0$ for some $x \in X_0$, then for small enough $\varepsilon > 0$, εx is in an ε-neighborhood of 0. Now, $\phi(\varepsilon x,0) = \varepsilon c^tx < 0 = \phi(0,0)$. This contradicts the fact that $(0,0)$ is a local minimum. Hence $c^tx \geq 0$ and likewise $(d^ty) \geq 0$.

Now suppose $x^tCy < 0$, $c^tx = 0$ and $d^ty = 0$ for some $x \in X_0$ and $y \in Y_0$. For small enough $\varepsilon > 0$, $(\varepsilon x,\varepsilon y)$ is in an ε-neighborhood of $(0,0)$. $\phi(\varepsilon x,\varepsilon y) = \varepsilon c^tx + \varepsilon d^ty + \varepsilon^2 x^tCy < 0 = \phi(0,0)$ which contradicts the fact that $(0,0)$ is a local minimum. Hence $x^tCy < 0$ implies either $c^tx > 0$ or $d^ty > 0$.

<u>Corollary 3.24</u>. Let $(0,0)$ be a nondegenerate vertex of BLP 3. Then it is a local minimum if and only if

i. $c \geq 0$ and $d \geq 0$ and

ii. either $c_i > 0$ or $d_j > 0$ whenever $c_{ij} < 0$.

Proof. We will show that conditions (i) and (ii) of Theorem 3.23 and Corollary 3.24 are equivalent when $(0,0)$ is a nondegenerate vertex.

i. $c \geq 0$ and $d \geq 0$ implies that $c^tx \geq 0$ and $d^ty \geq 0$ since $x \geq 0$ and $y \geq 0$ for each $x \in X_0$ and $y \in Y_0$. Conversely, since 0 is a nondegenerate vertex of X_0, there exists a point $x^k = (0,0,\ldots,x_k,0,\ldots,0)^t$ such that $x_k > 0$, $k = 1,\ldots,m$. Then $c^tx^k \geq 0$ implies $c_k \geq 0$, $k = 1,\ldots,m$. Hence $c \geq 0$. Similarly $d \geq 0$.

ii. Let $c_i > 0$ or $d_j > 0$ whenever $c_{ij} < 0$. Suppose that for some $\hat{x} \in X_0$, $\hat{y} \in Y_0$, we have $\hat{x}^tC\hat{y} < 0$. Since $\hat{x}^tC\hat{y} < 0$, there must exist some

$c_{ij} < 0$, $\hat{x}_i > 0$, $\hat{y}_j > 0$. From hypothesis, $c_{ij} < 0$ implies either $c_i > 0$ or $d_j > 0$. Hence either $c_i\hat{x}_i > 0$ or $d_j\hat{y}_j > 0$. From condition (i), $c \geq 0$ and $d \geq 0$. Since $\hat{x} \in X_0$, $\hat{y} \in Y_0$, $\hat{x} \geq 0$ and $\hat{y} \geq 0$. Now $c^t\hat{x} \geq c_i\hat{x}_i$ and $d^t\hat{y} \geq d_j\hat{y}_j$. Hence either $c^t\hat{x} > 0$ or $d^t\hat{y} > 0$.

Conversely, suppose $x^tCy < 0$ implies either $c^tx > 0$ or $d^ty > 0$. Let some $c_{ij} < 0$. If $c_i = d_j = 0$, then, since (0,0) is non-degenerate, there exist $\hat{x} \in X_0$, $\hat{y} \in Y_0$ with $\hat{x} = (0,\ldots,\hat{x}_i,\ldots,0)$ and $\hat{y} = (0,\ldots,\hat{y}_j,\ldots,0)$ such that $\hat{x}_i > 0$ and $\hat{y}_j > 0$. But $\hat{x}^tC\hat{y} < 0$ and $c^t\hat{x} = d^t\hat{y} = 0$ contradicting our hypothesis.

<u>Corollary 3.25</u>. (0,0) is a local minimum of BLP 3 if

i. $c \geq 0$ and $d \geq 0$.

ii. either $c_i > 0$ or $d_j > 0$ whenever $c_{ij} < 0$.

Proof. Let $x \in X_0$, $y \in Y_0$. This implies $x \geq 0$, $y \geq 0$. Then we can show that conditions (i) and (ii) of this Corollary imply conditions (i) and (ii) of Theorem 3.23. The proof is exactly the same as for Corollary 3.24.

Konno [26] has proved Corollary 3.22, 3.24 and 3.25. Later in this chapter, we will give an algorithm for finding an initial point of BLP 1 which, besides being a local star minimum and a local minimum, also possesses some special properties. We will show that the point is a local minimum by using Theorem 3.21 and 3.23. Corollary 3.22 and 3.25 are not adequate for this purpose in the case of degeneracy. The initial point is used as a starting point for the algorithms of Chapter IV and V. We will now characterize the properties of a local star minimum and a strong local star minimum, which is defined below.

Definition 3.26. A *strong local star minimum* is a local star minimum which is also a local minimum.

Suppose $(\bar{x},\bar{y})$ is a local star minimum of BLP 1. Let us transform the origin of the coordinate system to $(\bar{x},\bar{y})$ and obtain a problem of the form BLP 3, with $(0,0)$ as a local star minimum of BLP 3.

Theorem 3.27. The origin $(x,y) = (0,0)$ is a local star minimum of BLP 3 if and only if $c^t x \geq 0$ and $d^t y \geq 0$ for each $x \in X_0$ and $y \in Y_0$.

Proof. Suppose $(0,0)$ is a local star minimum. From Lemma 3.19, $x = 0$ is a solution to the problem:

$$\text{Min} \quad \phi(x,0) = c^t x, \quad x \in X_0$$

Hence $c^t x \geq 0$ for each $x \in X_0$. Likewise, $d^t y \geq 0$ for each $y \in Y_0$.

To prove the converse, suppose $c^t x \geq 0$ and $d^t y \geq 0$ for each $x \in X_0$ and $y \in Y_0$. Suppose $(0,0)$ is not a local star minimum. Consider the problems:

$$\text{P12:} \quad \text{Min } \phi(x,0) = c^t x \qquad \text{P13:} \quad \text{Min } \phi(0,y) = d^t y$$
$$x \in X_0 \qquad\qquad\qquad\qquad y \in Y_0$$

Then, from Lemma 3.19, either 0 is not a solution to P12 or 0 is not a solution to P13. In the former case, if $\hat{x}$ is a solution, then $c^t\hat{x} < \phi(0,0) = 0$, which contradicts the hypothesis that $c^t x \geq 0$ for each

$x \in X_0$. There is a similar contradiction if $y = 0$ is not a solution to P13.

Corollary 3.28. Let (0,0) be a nondegenerate vertex of BLP 3. Then it is a local star minimum if and only if $c \geq 0$ and $d \geq 0$.

Proof. Since the origin is a nondegenerate vertex of X_0, there exist extreme points $x^k = (0,0,\ldots,x_k,0,\ldots,0)^t$, $k = 1,\ldots,m$ with $x_k > 0$. From Theorem 3.27, $c^t x^k \geq 0$ and hence $c_k \geq 0$ for $k = 1,\ldots,m$. Likewise $d \geq 0$.

The converse follows from Theorem 3.27 by observing that $x \geq 0$ and $y \geq 0$ for each $x \in X_0$, $y \in Y_0$.

Corollary 3.29. The origin (0,0) is a local star minimum of BLP 3 if $c \geq 0$ and $d \geq 0$.

Proof. Follows from Theorem 3.27 by observing that $x \geq 0$ and $y \geq 0$ for each $x \in X_0$, $y \in Y_0$.

Theorem 3.30. Let (0,0) be a local minimum of BLP 3. Then it is a local star minimum, and hence a strong local star minimum, of BLP 3.

Proof. Suppose (0,0) is not a local star minimum. Then either there exists an adjacent extreme point x^p to the origin or an adjacent extreme point y^q to the origin such that either $\phi(x^p,0) < \phi(0,0) = 0$ or $\phi(0,y^q) < \phi(0,0) = 0$. Let us consider the former case. The proof for the latter is similar. $\phi(x^p,0) = c^t x^p < 0$. For small enough $\varepsilon > 0$, the point $(\varepsilon x^p,0)$ is in an ε-neighborhood of (0,0). Now

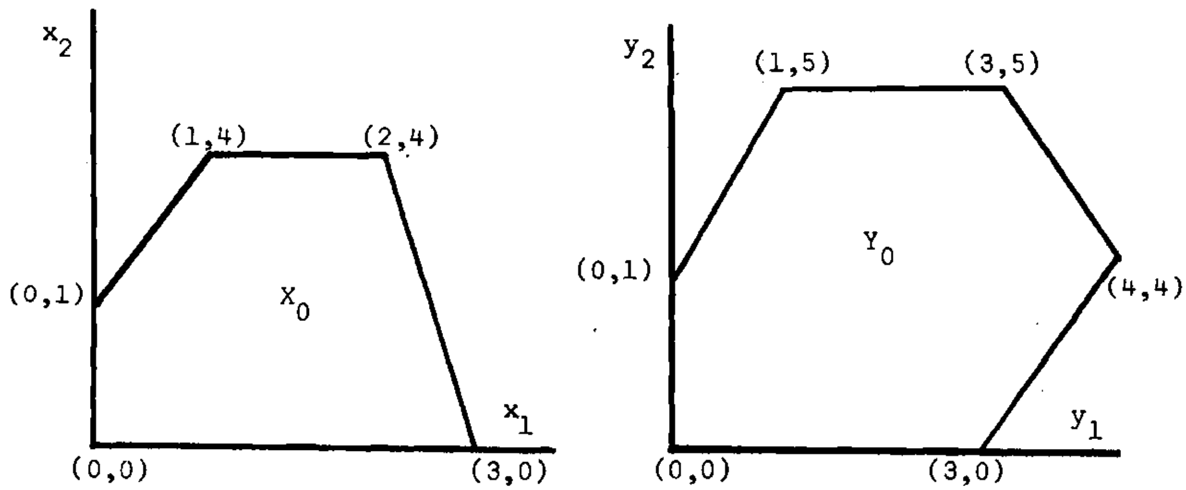
$\phi(\varepsilon x^p, 0) = \varepsilon c^t x^p < 0 = \phi(0,0)$. This contradicts the fact that $(0,0)$ is a local minimum.

We have been implicitly assuming that a local star minimum need not be a local minimum. That this is indeed true is shown by the following example.

Example Problem 2

$$\text{Minimize} \quad \phi(x,y) = x_2 + y_1 + x_2 y_1 - x_1 y_2 + x_2 y_2$$

$$\text{Subject to} \quad (x_1, x_2) \in X_0,\ (y_1, y_2) \in Y_0.$$



The origin is a local star minimum, with $\phi(0,0) = 0$. Let us consider the point $\bar{x} = (0.1, 0)^t$, $\bar{y} = (0, 0.1)^t$. $\phi(\bar{x}, \bar{y}) = -0.01 < 0 = \phi(0,0)$. Hence it is not a local minimum. The point $x = (3,0)^t$, $y = (1,5)^t$ is a strong local star minimum.

For the purposes of our cutting plane algorithm, we need a point that possesses certain special properties. We will show why these properties are required in the next chapter. We have called such a point a *pseudo-global minimum*. We will first state these properties, then show that a pseudo-global minimum is a strong local star minimum and give an algorithm for finding a pseudo-global minimum.

Definition 3.31. Let (0,0) be a local star minimum of BLP 3. Suppose each extreme point y^i of Y_0 for which $d^ty^i = 0$ satisfies: $\phi(x,y^i) \geq \phi(0,0)$ for each $x \in X_0$. Then (0,0) is called a *pseudo-global minimum*.

Let $y^1,y^2,\ldots,y^k$ be extreme points of Y_0 satisfying the property $d^ty^i = 0$ and $\phi(x,y^i) \geq \phi(0,0)$ for $i = 1,\ldots,k$. We can see that for each $x \in X_0$ and $y \in \text{conv}[0,y^1,y^2,\ldots,y^k]$ (0,0) is the global minimum, where $\text{conv}[0,y^1,y^2,\ldots,y^k]$ is the convex hull of the points $0,y^1,y^2,\ldots,y^k$. This is the reason why we have called such a point a pseudo-global minimum. We will show that a pseudo-global minimum is a strong local star minimum, and hence a local minimum.

Theorem 3.32. Let (0,0) be a pseudo-global minimum. Then (0,0) is a strong local star minimum.

Proof. Since (0,0) is a local star minimum, from Theorem 3.27, $c^tx \geq 0$ and $d^ty \geq 0$ for each $x \in X_0$ and $y \in Y_0$. From Theorem 3.23, we only need to show that if $\hat{x}^tC\hat{y} < 0$ for some $\hat{x} \in X_0$, $\hat{y} \in Y_0$, then $c^t\hat{x} > 0$ or $d^t\hat{y} > 0$. Consider a $\hat{x} \in X_0$, $\hat{y} \in Y_0$ satisfying $\hat{x}^tC\hat{y} < 0$. Now $d^t\hat{y} \geq 0$. If $d^t\hat{y} > 0$, the theorem stands proved. Hence we will consider the case when $d^t\hat{y} = 0$.

Since $\hat{y} \in Y_0$, it can be expressed as $\hat{y} = \sum_{i=1}^{s} \lambda_i y^i$, $\lambda_i > 0$,

$\sum_{i=1}^{s} \lambda_i = 1$, where y^i, $i = 1,\ldots,s$ are extreme points of Y_0. Hence, $\hat{x}^t C\hat{y} = \sum_{i=1}^{s} \lambda_i \hat{x}^t Cy^i < 0$ implies that for some $i = r$, $\hat{x}^t Cy^r < 0$. Also, $d^t\hat{y} = \sum_{i=1}^{s} \lambda_i d^t y^i = 0$ implies $d^t y^i = 0$, $i = 1,\ldots,s$, since $\lambda_i > 0$ and $d^t y^i \geq 0$ from Theorem 3.27. In particular, $d^t y^r = 0$. Since (0,0) is a pseudo-global minimum and $d^t y^r = 0$, $\phi(x,y^r) = c^t x + d^t y^r + x^t Cy^r \geq \phi(0,0)$ for each $x \in X_0$. Letting $x = \hat{x}$ and noting that $d^t y^r = 0$, we get $c^t\hat{x} > -\hat{x}^t Cy^r > 0$ since we have shown that $\hat{x}^t Cy^r < 0$. Hence (0,0) is a local minimum and consequently a strong local star minimum.

In the next section, we state an algorithm for finding a pseudo-global minimum.

3.5 Determination of Pseudo-Global Minimum

The following algorithm will be used to determine the pseudo-global minimum $(\bar{x},\bar{y})$ that will form the starting point for the global optimal algorithms of Section 4.5 and 5.5. For the ε-optimal algorithm (Section 4.8) the local star minimum $(\bar{x},\bar{y})$ obtained at the end of step (2) is adequate. However, since step (3) could improve the objective function with comparatively less computations, it is included in the ε-optimal algorithm also.

Algorithm A

1. If X_0 is empty, terminate. Otherwise find a feasible extreme point x^1 of X_0. This can be done, for example, with the help of a Phase I procedure.

2. Solve the linear program P14 to get an extreme point y^1.

$$\text{P14:}\quad \text{Minimize} \quad c^t x^1 + d^t y + (x^1)^t Cy$$

$$\text{Subject to} \quad Fy \le f, \quad y \ge 0$$

Solve the linear program P15 to get an extreme point x^2

$$\text{P15:}\quad \text{Minimize} \quad c^t x + d^t y^1 + x^t Cy^1$$

$$\text{Subject to} \quad Ex \le e, \quad x \ge 0$$

Alternate between P14 and P15, each time using the new extreme point generated till the procedure converges to the extreme points $\bar{x}$ and $\bar{y}$.

3. If $\bar{y}$ is the unique optimal solution corresponding to $x = \bar{x}$, then terminate, $(\bar{x},\bar{y})$ is the pseudo-global minimum. Otherwise generate the alternative optimal solutions $y^1,y^2,\ldots,y^k$, that is $c^t\bar{x} + d^t y^i + \bar{x}^t Cy^i = c^t\bar{x} + d^t\bar{y} + \bar{x}^t C\bar{y}$ for $i = 1,\ldots,k$. Solve problem P15 for each of $y = y^i$ and let the corresponding solutions be x^i. Let $c^t x^r + d^t y^r + (x^r)^t Cy^r = \min\{c^t x + d^t y + x^t Cy$ for $(x,y)=(\bar{x},\bar{y})$ and $(x,y)=(x^i,y^i)$, $i=1,\ldots,k\}$.

If $x^r = \bar{x}$, terminate and $(\bar{x},\bar{y})$ is a pseudo-global minimum. Otherwise, go to step (2) with $x = x^r$.

Finiteness of the algorithm is assured because the number of extreme points is finite, the objective function value has a finite minimum, and each sequence of steps from (2) through (3) yields a strict decrease in the value of the objective function.

The algorithm terminates at a point $(\bar{x},\bar{y})$ obtained at step (2) and, from Lemma 3.19, this point is a local star minimum. Consider the problem BLP 3 obtained by transferring the origin to $(\bar{x},\bar{y})$. From step (3) if $c^t\bar{x} + d^ty^i + \bar{x}^tCy^i = c^t\bar{x} + d^t\bar{y} + \bar{x}^tC\bar{y}$, since $\bar{x} = \bar{y} = 0$, we get $d^ty^i = 0$. Also from step (3), $\phi(x,y^i) \geq \phi(x^i,y^i) \geq \phi(0,0)$ for each $x \in X_0$, and each y^i such that $d^ty^i = 0$, since x^i was obtained by solving the problem: Minimize $\phi(x,y^i)$, $x \in X_0$. Hence $(\bar{x},\bar{y})$ by Definition 3.31 is a pseudo-global minimum.

3.6 Determination of a Good Feasible Solution

Though the point $\bar{x}$ determined above will suffice for the purposes of our algorithms, it is felt that a better starting solution will tend to reduce the total amount of computations that will have to be done in order to find the global minimum. Thus, any reasonable extra effort expended at this stage will, perhaps, be worth while. In some practical cases, we may be interested in obtaining a reasonably good solution without regard to global optimality. With the help of the bounds on the objective function value of BLP 1 that we will generate in Chapter VI, we may even be able to determine how good a solution we actually have. We now indicate some ways of obtaining such a solution for these two purposes.

Having determined the point $(\bar{x},\bar{y})$ as above, once again we transform the origin to $(\bar{x},\bar{y})$ and obtain a problem of the form BLP 3 with the additional property that $c \geq 0$ and $d \geq 0$. Let us define the set $T = \{(i,j) | c_{ij}<0,\ i=1,\ldots,m;\ j=1,\ldots,n\}$. If T is an empty set, then the origin is a global minimum, so let $T \neq \phi$. Let

$\phi_{ij}(x_i,y_j) = c_ix_i + d_jy_j + c_{ij}x_iy_j$ for all $(i,j) \in T$. We would now like to examine the extreme points adjacent to the origin of the x and y variables. These are of the form

$$x^i = (0,0,\ldots,x_i,\ldots,0)$$

where x_i is given by

$$x_i = \min\ \{\frac{e_k}{e_{ki}},\ e_{ki}>0 \quad \text{for all } k\}$$

and

$$y^j = (0,0,\ldots,y_j,\ldots,0)$$

with y_j given by

$$y_j = \min\ \{\frac{f_k}{f_{kj}},\ f_{kj}>0 \quad \text{for all } k\}$$

where e_k, e_{ki}, f_k, f_{kj} are the entries in the final simplex tableau in the x and y variables.

In particular, we now evaluate the effect of making pairs of changes by moving simultaneously to x^i and y^j with $(i,j) \in T$. This can be done by simply evaluating $\phi_{ij}(x_i,y_j)$ for all $(i,j) \in T$ with x_i and y_j defined by the expressions given above. If $\min\{\phi_{ij}(x_i,y_j),\ (i,j)\in T\} \geq 0$ no improvement occurs by making such a movement. Otherwise, we will have found a new extreme point (x^r,y^s) with a strict decrease in the value of the objective function as compared with $(\bar{x},\bar{y})$. In this case we can go back to step (1) of our algorithm with $x = x^r$. If no improvement occurs, since we already have available the extreme points x^i and y^j adjacent to the origin, we can solve a series of linear programs over Y_0 with x fixed in turn at each x^i and over X_0 with y fixed in turn at each y^j. If any of these problems yield a strict decrease in objective function value, once again we go to step (1) of our algorithm. If not, we can terminate the search for a solution at this stage. These steps are now summarized below.

Algorithm B

1. Determine the pseudo-global minimum $(\bar{x},\bar{y})$. Transform the origin to $(\bar{x},\bar{y})$. Determine the extreme points adjacent to the new origin as follows:

$$x^i = (0,0,\ldots,x_i,\ldots,0), \quad x_i = \min\{\frac{e_k}{e_{ki}},\ e_{ki}>0 \text{ for all } k\}$$

$$y^j = (0,0,\ldots,y_j,\ldots,0), \quad y_j = \min\{\frac{f_k}{f_{kj}},\ f_{kj}>0 \text{ for all } k\}$$

Define $T = \{(i,j)|c_{ij}<0,\ i=1,\ldots,m;\ j=1,\ldots,n\}$ and $\phi_{ij}(x_i,y_j) = c_ix_i + d_jy_j + c_{ij}x_iy_j + k$ for all $(i,j) \in T$, with x_i and y_j defined as above.

2. If $T = \phi$, terminate since the global minimum has been found. If $\phi_{ij}(x_i,y_j) \geq k$ for all (i,j) go to (3); otherwise, let (x^r,y^s) be an extreme point such that $\phi_{ij}(x_r,y_s) < k$. Go to (1) with initial solution $x = x^r$.

3. With x fixed at each of x^i solve the following problem:

$$\text{Minimize} \quad \phi(x^i, y),$$

$$\text{Subject to} \quad y \in Y_0$$

If the optimal value of the objective function is less than the best available for some i, go to (1) with $y = y^{i*}$, where y^{i*} is the optimal solution to the *ith* problem. Otherwise terminate; $(\bar{x},\bar{y})$ is a good feasible solution.

3.7 Summary

In this chapter, we have shown that the optimal solution to the Bilinear Programming problem will be attained at an extreme point, but there may exist local star minima different from the global minimum. We have then shown how to find a pseudo-global minimum, which is used as a starting point for the two algorithms that we will be developing later on. Finally, we have shown how to obtain a good feasible solution.

CHAPTER IV

CUTTING PLANE ALGORITHM

4.1 Introduction

In this chapter, we will be developing a cutting plane algorithm for solving the Bilinear problem. Cutting plane algorithms have been used in integer and nonlinear programming. Gomory [20] first showed how to solve an integer linear program by treating the problem as a linear programming problem and adding a series of cuts till an integer solution is obtained. Since that time, a variety of cutting plane algorithms have been developed with different objectives in mind. Ease of generation of the cutting plane was one such objective, which led to an inequality of the type $\sum_{j \in J} x_j \geq 1$, where J is the set of nonbasic variables. Round off problems on computers led to the development of dual all-integer cuts in which each tableau is all integer. The desirability of having a feasible integer solution throughout the cut generation phase motivated the development of primal all-integer cutting plane algorithms. Finally, a number of algorithms (see, for example, [30,6,2] were developed with the objective of cutting off as much of the feasible noninteger region as possible. In fact, the objective behind most of the cutting plane algorithms that have been recently published seems to be to generate deeper cuts. Computational experience (see [19]) indicates that cutting plane algorithms are not, in general, an effective solution procedure for realistic integer programming problems.

A number of very efficient algorithms have been developed for solving integer programs with special structures. In such cases, cutting plane algorithms do very poorly because they do not depend on the problem structure. Moreover, they tend to destroy the problem structure. However, for general problems, these methods may still be competitive because of their relative simplicity. However, no extensive comparisons have been made for such problems. Thus, generation of deep cuts is a worthwhile research effort for solving more efficiently integer programming problems of a general nature.

In nonlinear convex programming, cutting plane algorithms have been used to extend the power of linear programming to the more general class of problems. In the concave-cutting plane algorithm [48], a problem of the form: Maximize $f(x)$, subject to $g_i(x) \geq 0$, $i = 1,\ldots,m$, where f and g_i are concave is transformed to a problem of the form: Maximize $q^t y$, subject to $h(y) \geq 0$ with a linear objective function and one constraint involving a concave function. The feasible set of this problem is assumed to be a subset of $U = \{y \mid Ay \leq b\}$ with A and b known. If the optimal solution of the problem: Maximize $q^t y$, subject to $Ay \leq b$ satisfies the constraint $h(y) \geq 0$, the original problem is solved. Otherwise a constraint is added to the linear programming problem which cuts off the current solution but no feasible solution to $h(y) \geq 0$.

It is also shown in [48] that generalized linear programming as applied to a problem with a nonlinear objective function and constraints (see [28]) is nothing but a cutting plane approach. Other applications of cutting plane methods to nonlinear programming are discussed in [48].

In the general area of nonconvex programming, the strengths and weaknesses of different approaches have not been firmly established. Only problems with special structures have been solved and very few computational results have been published. There are thus no guidelines to follow. We have decided on a cutting plane approach to the Bilinear problem since that indicates promise of yielding a better capability for solving realistic problems. We will generate deep cuts so as to exhaust the feasible region as quickly as possible. Moreover, we will maintain the special structure of the problem. Unlike the cutting plane algorithms of integer and convex programming, it is usually very hard to prove mathematically convergence of such algorithms in the nonconvex case. Several of the current procedures available for solving nonconvex quadratic programs are either proven nonconvergent or do not have any proof at all. Only two algorithms prove that they will converge finitely to an ε-optimal solution, which is a point at which the objective function value is no more than ε greater than the true global minimum. We will develop two algorithms and will prove infinite convergence of our global optimum algorithm and also develop a finitely convergent ε-optimal solution procedure. We will show that our cuts are deeper than those generated by Konno [26], hence convergence can be expected to be faster.

The basis of the cutting plane to be developed in this chapter is the theory of polaroids as developed by Burdet [7]. Polaroids have been used by Balas [2] as a totally new approach for generating a cutting plane for an integer programming problem. As indicated earlier, they have been used in [4] and [3] for solving convex and nonconvex quadratic programming problems.

4.2 Generalized Polar Sets

The polar set A^* of a set $A \subset R^n$ has been defined in the literature (see, for example [21]) as

$$A^* = \{x \in R^n \mid \langle x,y \rangle \leq 1 \quad \text{for all } y \in A\}.$$

This has been generalized in [7] to define what is called the generalized polar of a set.

Let $A \subset R^m$. Given a function $f\colon R^n \times R^m \to E^1$, and a scalar k, the generalized polar of A is defined as

$$A^0(k) = \{x \in R^n \mid f(x,y) \leq k \quad \text{for all } y \in A\}.$$

The generalization consists in associating a numerical function with the set A and in changing the right-hand side of the inequality from 1 to an arbitrary number. The objective behind doing this is to construct polar sets with respect to the objective function of a nonconvex optimization problem using as the scalar k the current global optimum of the problem and to find the intersection of the polar with as large a subset of the feasible region as is computationally tractable. The optimum over this subset would then be known. If the entire feasible region can be broken up into a finite number of such subsets, the problem will have been solved. We first present some definitions and properties of generalized polars as applied to Bilinear Programming problems.

We restate our problem BLP 1 as:

$$\text{Minimize} \quad \phi(x,y) = c^t x + d^t y + x^t C y$$

$$\text{Subject to} \quad x \in X_0 = \{x \in R^m \mid Ex \le e,\ x \ge 0\}$$
$$y \in Y_0 = \{y \in R^n \mid Fy \le f,\ y \ge 0\}$$

where X_0 and Y_0 are bounded.

Definition 4.1. The generalized polar of Y_0 for a given scalar k is defined as

$$Y^0(k) = \{x \mid c^t x + d^t y + x^t C y \ge k \quad \text{for all } y \in Y_0\}$$

The generalized polar $Y^0(k)$ has some very useful properties. These properties are defined by the following lemmas.

Lemma 4.2. $Y^0(k)$ is a polyhedral set, and hence a convex set.

Proof. Let $V = \{y^i \mid y^i$ is an extreme point of $Y_0\}$. Since Y_0 is a bounded polytope, V is a finite set and any $y \in Y_0$ can be expressed as a convex combination of the elements of V. Let $\hat{Y}(k) = \{x \mid c^t x + d^t y^i + x^t C y^i \ge k$ for each $y^i \in V\}$. For any given $y^r \in V$, $\{x \mid c^t x + d^t y^r + x^t C y^r \ge k\}$ is a closed halfspace. Thus $\hat{Y}(k)$ is a polyhedral set, being the intersection of a finite number of closed halfspaces. We will show that

$\hat{Y}(k) = Y^0(k)$. Let $\bar{x} \in \hat{Y}(k)$. Then $c^t\bar{x} + d^t y^i + \bar{x}^t C y^i \geq k$ for all $y^i \in V$. For $\lambda_i \geq 0$ and $\sum \lambda_i = 1$ we have

$$\sum_{y^i \in V} \lambda_i (c^t\bar{x} + d^t y^i + \bar{x}^t C y^i) \geq k$$

or

$$c^t\bar{x} + d^t \sum_{y^i \in V} \lambda_i y^i + \bar{x}^t C \sum_{y^i \in V} y^i \geq k$$

or

$$c^t\bar{x} + d^t y + \bar{x}^t C y \geq k \quad \text{for all } y \in Y_0$$

Hence $\bar{x} \in Y^0(k)$.

Conversely, if $\bar{x} \in Y^0(k)$, then

$$c^t\bar{x} + d^t y + \bar{x}^t C y \geq k \quad \text{for all } y \in Y_0,$$

and hence in particular for all the extreme points of Y_0.

Hence $c^t\bar{x} + d^t y^i + \bar{x}^t C y^i \geq k$ for all $y^i \in V$.

Hence $\bar{x} \in \hat{Y}(k)$ so that $\hat{Y}(k) = Y^0(k)$.

Lemma 4.3. $Y^0(k)$ contains no point $x \in X_0$ such that $c^t x + d^t y + x^t C y < k$ for some $y \in Y_0$.

This follows directly from the definition of $Y^0(k)$.

Lemma 4.4. $(\bar{x},\bar{y})$ solves BLP 1 if and only if $X_0 \subset Y^0(k)$ for $k = c^t\bar{x} + d^t\bar{y} + \bar{x}^t C\bar{y}$, where $\bar{x} \in X_0$ and $\bar{y} \in Y_0$.

Proof. Let $(\bar{x},\bar{y})$ be optimal to BLP 1, and let $\hat{x} \in X_0$. Then $c^t\hat{x} + d^ty + x^tCy \geq c^t\bar{x} + d^t\bar{y} + \bar{x}^tC\bar{y} = k$ for all $y \in Y_0$. Hence $\hat{x} \in Y^0(k)$ so that $X_0 \subset Y^0(k)$.

Conversely, if $X_0 \subset Y^0(k)$, then by definition of $Y^0(k)$, there is no point $x \in X_0$ such that $c^tx + d^ty + x^tCy < k$ for some $y \in Y_0$. Hence $(\bar{x},\bar{y})$ is optimal.

4.3 Generation of Polar Cuts

We now come to the question of generating a cutting plane to eliminate part of the feasible region. In order to maintain the separable structure of the Bilinear problem, the cut will be defined in terms of either x and u or y and v. In case one of the sets X_0 or Y_0 has a special structure, for example a transportation structure as in a transportation-location problem, we would like to preserve that structure and hence generate cutting planes with respect to the other set only. Our description will be for generating a cut in the variables x and u.

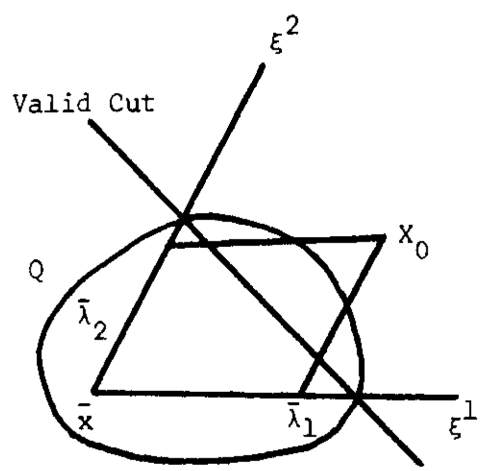


Figure 2. Example of Valid Polar Cut

The basic ideas of the method are as follows. Let $(\bar{x},\bar{y})$ be a pseudo-global minimum (see Figure 2). Let us consider the polyhedral cone C with vertex at $\bar{x}$ and whose m extreme rays are given by: $\xi^j = \{x \mid x=\bar{x}-\bar{e}^j\lambda_j,\ \lambda_j \geq 0\}$, $j \in J$, where the vectors $\bar{e}^j$ are obtained as described in Section 3.3. Suppose $Q \subset R^m$ is any closed convex set that has the property that $\phi(x,y) \geq \phi(\bar{x},\bar{y}) \geq k$ for each $x \in Q$ and $y \in Y_0$. Since $(\bar{x},\bar{y})$ is a pseudo-global and hence a local star minimum, $\phi(\bar{x},y) \geq \phi(\bar{x},\bar{y})$ for each $y \in Y_0$. Hence $\bar{x} \in Q$. Let us consider the set $C \cap Q$. Since $\phi(x,y) \geq \phi(\bar{x},\bar{y})$ for each $x \in C \cap Q$ and $y \in Y_0$, the global minimum over $C \cap Q$ is known. In general, it may not be very easy to define the set $C \cap Q$, but a special subset of it is easier to construct algebraically. Suppose we can find m points $x^1,\ldots,x^m$ $(x^i \neq \bar{x})$ which are the intersection of the m rays ξ^j with the boundary of Q. Let H be the unique hyperplane passing through the points. If H^- is the closed halfspace containing $\bar{x}$, then $C \cap H^- \subset C \cap Q$. If the set Q is suitably defined finding the points $x^1,\ldots,x^m$ is not too difficult.

Thus the global minimum over $C \cap H^-$ is known, so that H^+ will be a valid cutting plane, where H^+ is the closed halfspace not containing $\bar{x}$. If $X_0 \subset C \cap H^-$, the Bilinear problem is solved, otherwise a new pseudo-global minimum is found, and the procedure is repeated until X_0 is exhausted. It is intuitively clear that the larger the set Q, the deeper will be the cut generated, and that is where polar sets become useful.

For the rest of the chapter, whenever we refer to the set X_0, we mean the feasible set after the inequalities (cuts), if any, are added. Note that these cuts would still maintain the form of BLP 3 since X_0 will still be a bounded polytope. All the properties developed in Chapter III will hold for this X_0.

Let $(\bar{x},\bar{y})$ be a pseudo-global minimum of BLP 1 and let $\bar{p}^t = (\bar{x}^t\bar{u}^t)$. Suppose the current best value of the objective function $\phi(x,y)$ of BLP 1 is k, which may or may not be equal to $\phi(\bar{x},\bar{y})$.

Definition 4.5. Given m positive scalars $\bar{\lambda}_1,\bar{\lambda}_2,\ldots,\bar{\lambda}_m$, then the inequality: $\sum_{j\in J} p_j/\bar{\lambda}_j \geq 1$ *is a valid cutting plane* with respect to $\bar{p}^t = (\bar{x}^t\bar{u}^t)$ if $\sum_{j\in J} \bar{p}_j/\bar{\lambda}_j < 1$ but $\sum_{j\in J} p_j/\bar{\lambda}_j \geq 1$ for all $p \in P = \{(x^t u^t)^t | Ex+u=e,$ $x\geq 0,\ u\geq 0\}$, such that $c^t x + d^t y + x^t Cy < k$ for some $y \in Y_0$.

A valid cutting plane is one which cuts off the pseudo-global minimum but does not cut off any feasible point that may have an objective function value smaller than k for some $y \in Y_0$. The following theorem specifies the properties that the set Q must have and how to define a valid cutting plane for the Bilinear problem.

Suppose $\bar{x}$ is an extreme point of X_0 satisfying the conditions stated in Theorem 4.6 below. The particular extreme point $\bar{x}$ that we will be using for our algorithm is such that $(\bar{x},\bar{y})$ is a pseudo-global minimum of BLP 1. Let the m rays of X_0' incident on $\bar{x}$ be $\xi^j = \{x | x = \bar{x} - \bar{e}^j\lambda_j,\ \lambda_j \geq 0\}$, $j \in J$ where the $\bar{e}^j$ are obtained from Equation (6) of Chapter III. Suppose Q is any closed convex set. If a ray ξ^j is *entirely* contained in Q, we will write $\xi^j \subset Q$. If only part of the ray is in Q, we will write $\xi^j \not\subset Q$. The relative interior of a ray ξ^j will be denoted by $r.i.(\xi^j)$.

<u>Theorem 4.6.</u> Let $\bar{x}$ be an extreme point of X_0. Let Q be a closed convex set such that

(i) $\bar{x} \in Q$.

(ii) $Q \cap r.i.(\xi^j) \neq \phi$ for each $j \in J$.

(iii) $Q \cap \{x \in X_0 | c^t x + d^t y + x^t C y < k$ for some $y \in Y_0\} = \phi$.

Let $\bar{\lambda}_j$ be defined by:

$$\bar{\lambda}_j = \max[\lambda_j | \bar{x} - \bar{e}^j \lambda_j \in Q] \quad \text{if } \xi^j \not\subset Q$$
$$= \infty \quad \text{if } \xi^j \subset Q \quad \text{for all } \lambda_j \geq 0.$$

Then the inequality: $\sum_{j \in J} p_j / \bar{\lambda}_j \geq 1$ is a valid cutting plane.

Proof. Since $Q \cap r.i.(\xi^j) \neq \phi$, there exists a point $x = (\bar{x} - \bar{e}^j \lambda_j) \in Q$ with $\lambda_j > 0$ for each $j \in J$. Hence $\bar{\lambda}_j > 0$, i.e., $0 \leq 1/\bar{\lambda}_j < \infty$ for each $j \in J$. Since the value of the nonbasic variables at $(\bar{x}, \bar{u})$ is zero (i.e., $\bar{p} = 0$, $j \in J$), we have $\sum_{j \in J} \bar{p}_j / \bar{\lambda}_j < 1$. From Definition 4.5, all that remains to show is that $\sum_{j \in J} p_j / \bar{\lambda}_j \geq 1$ for each $p \in P$ such that $c^t x + d^t y + x^t C y < k$ for some $y \in Y_0$.

Let $S = \{x = \bar{x} - \sum_{j \in J} \bar{e}^j p_j,\ \sum_{j \in J} p_j / \bar{\lambda}_j \leq 1,\ p_j \geq 0,\ j \in J\}$.

Consider the halflines ξ^j, $j \in J$ which are the m rays emanating from the extreme point $\bar{x}$ of S. S is a subset of the cone with vertex $\bar{x}$ and with generators ξ^j. Hence any point $x \in S$ can be expressed as a convex combination of points $(\bar{x} - \bar{e}^j \lambda_j)$ on ray ξ^j, $\lambda_j \geq 0$, $j \in J$.

$$x = \sum_{j \in J} \mu_j(\bar{x}-\bar{e}^j\lambda_j), \ \mu_j \geq 0, \ \sum_{j \in J} \mu_j = 1$$

$$= \bar{x} - \sum_{j \in J} \bar{e}^j \mu_j \lambda_j$$

Letting $p_j = \mu_j\lambda_j$, we have $p_j \geq 0$, since $\mu_j \geq 0$ and $\lambda_j \geq 0$. Also $\sum_{j \in J} p_j/\bar{\lambda}_j = \sum_{j \in J} \mu_j\lambda_j/\bar{\lambda}_j \leq 1$ for $x \in S$. For points on the rays ξ^j, $\mu_j = 1$, $j \in J$. Hence $\lambda_j/\bar{\lambda}_j \leq 1$ or $\lambda_j \leq \bar{\lambda}_j$, $j \in J$. But $0 \leq \lambda_j \leq \bar{\lambda}_j$ implies $(\bar{x}-\bar{e}^j\lambda_j) \in Q$ from the definition of $\bar{\lambda}_j$. Hence we have expressed $x \in S$ as a convex combination of points in Q. Since Q is convex, this implies that $x \in Q$, that is $S \subset Q$.

Now let $S_1 = \{x \in X_0 | c^t x + d^t y + x^t Cy < k \text{ for some } y \in Y_0\}$. From condition (iii), $Q \cap S_1 = \phi$. Since $S \subset Q$, we have $S \cap S_1 = \phi$, i.e. corresponding to each $x \in S_1$, $\sum_{j \in J} p_j/\bar{\lambda}_j > 1$. Hence $\sum_{j \in J} p_j/\bar{\lambda}_j \geq 1$ is a valid cut.

In effect, this theorem will enable us to locate the global minimum of $\phi(x,y)$ over a subset of X_0 and over the entire region Y_0. By adding on the inequality corresponding to the valid cut to X_0, a new smaller feasible polytope for the x-variables is defined. By repeating this procedure enough number of times, we will attempt to partition X_0 into a number of regions, the global minimum over each being known. Let us now see how this can be done.

Consider the problem BLP 1. Suppose we have obtained a pseudo-global minimum $(\bar{x},\bar{y})$ as indicated in Chapter III. Let the current best value of $\phi(x,y)$ be k, which may or may not be attained at $(\bar{x},\bar{y})$. We would now like to show that a valid cut can be generated with the polar set $Y^0(k)$ playing the role of Q referred to in Theorem 4.6. We recall the definition of $Y^0(k)$:

$$Y^0(k) = \{x \mid \phi(x,y)=c^t x+d^t y+x^t Cy \geq k \text{ for each } y \in Y_0\}$$

<u>Lemma 4.7</u>. $Y^0(k)$ is a closed convex set having the properties:

(i) $\bar{x} \in Y^0(k)$.

(ii) $Y^0(k) \cap \text{r.i.}(\xi^j) \neq \phi$ for each $j \in J$.

(iii) $Y^0(k) \cap \{x \in X_0 \mid \phi(x,y) < k \text{ for some } y \in Y_0\} = \phi$

Proof. From Lemma 4.2, we see that $Y^0(k)$ is a convex polyhedral set. Hence it is a closed convex set. We now verify that $Y^0(k)$ does in fact possess the three properties specified.

(i) Since $(\bar{x},\bar{y})$ is a pseudo-global minimum and hence a local star minimum, we have $\phi(\bar{x},y) \geq \phi(\bar{x},\bar{y}) \geq k$ for each $y \in Y_0$. Hence $\bar{x} \in Y^0(k)$.

(ii) Let us transform the origin to $(\bar{x},\bar{y})$ and obtain a problem of the form BLP 3. Now $(0,0)$ is a pseudo-global minimum and hence a local star minimum of BLP 3. Hence from Lemma 3.19, $x = 0$ is an optimal solution to the problem:

$$\text{Minimize} \quad \phi(x,0) = c^t x + R$$

$$\text{Subject to} \quad x \in X_0$$

where R is the objective function value at $(\bar{x},\bar{y})$. Consider the set X_0' obtained from X_0 by deleting certain specific constraints. $X_0 \subset X_0'$.

From Theorem 3.20, the origin $x = 0$ is a vertex of X_0' and $\phi(x,0) = \phi(0,0)$ is a supporting hyperplane for X_0'. Hence $\phi(x,0) \geq \phi(0,0)$ for each $x \in X_0'$ that is, $c^t x + R \geq R$ or $c^t x \geq 0$ for each $x \in X_0'$. Also from Theorem 3.27, $d^t y \geq 0$ for each $y \in Y_0$.

Let V be the set of extreme points of Y_0. Consider any $x \in X_0'$, $y^i \in V$. For $0 \leq \varepsilon \leq 1$, $(\varepsilon x) \in X_0'$. Now $\phi(\varepsilon x, y^i) = \varepsilon c^t x + d^t y^i + \varepsilon x^t C y^i + R$. We have shown that $c^t x \geq 0$, $d^t y \geq 0$ for each $x \in X_0'$, $y \in Y_0$. If $d^t y^i > 0$, then there exists a small enough $\varepsilon > 0$ such that $\phi(\varepsilon x, y^i) \geq \phi(0,0) \geq k$ for each $x \in X_0'$.

Now suppose $d^t y^i = 0$. Then $\phi(0,y^i) = d^t y^i + R = R = \phi(0,0)$. Since $(0,0)$ is a pseudo-global minimum and $d^t y^i = 0$, from Definition 3.31, $\phi(x,y^i) \geq \phi(0,0)$ for each $x \in X_0$. Since $\phi(0,0) = \phi(0,y^i)$, $\phi(x,y^i) \geq \phi(0,y^i)$ for each $x \in X_0$. Consider the problem:

$$\text{Minimize} \quad \phi(x,y^i)$$

$$\text{Subject to} \quad x \in X_0$$

An optimal solution is $x = 0$. From Theorem 3.20, the origin $x = 0$ is a vertex of X_0' and $\phi(x,y^i) = \phi(0,y^i)$ is a supporting hyperplane for X_0'. Hence $\phi(x,y^i) \geq \phi(0,y^i) = \phi(0,0)$ for each $x \in X_0'$. Now if $x \in X_0'$, then $\varepsilon x \in X_0'$ for $0 \leq \varepsilon \leq 1$. Also $\phi(0,0) \geq k$ by hypothesis. Hence $\phi(\varepsilon x, y^i) \geq k$ for each $x \in X_0'$, $0 \leq \varepsilon \leq 1$.

We have thus proved that for each $y^i \in V$ and each $x \in X_0'$, $\phi(\varepsilon x, y^i) \geq k$ for small enough $\varepsilon > 0$. From Lemma 4.2, we know that $Y^0(k) = \hat{Y}(k) = \{x \,|\, \phi(x,y^i) \geq k \text{ for each } y^i \in V\}$. Hence $\varepsilon x \in Y^0(k)$ for small enough $\varepsilon > 0$. From Theorem 3.20, X_0' has precisely m edges incident on the vertex 0, and each ray ξ^j contains one such edge. By picking x on each edge, $x \neq 0$, $(\varepsilon x) \in$ r.i.(ξ^j) for small enough $\varepsilon > 0$ and for each $j \in J$. Hence $Y^0(k) \cap$ r.i.$(\xi^j) \neq \phi$ for each $j \in J$.

(iii) This is satisfied since from Lemma 4.3, $Y^0(k)$ contains no point $x \in X_0$ such that $\phi(x,y) < k$ for some $y \in Y_0$. Hence the lemma is proved.

We can set an upper bound on the ε determined in the proof of part (ii). Let us consider the following linear program:

$$\text{P16: Minimize} \quad x^t C y^i$$

$$\text{Subject to} \quad x \in X_0$$

for $y^i \in V$. Let L_i be the minimum value corresponding to extreme point y^i. Since X_0 is a bounded polytope, each L_i is finite if C is. Since Y_0 is a polytope, it has a finite number of extreme points. Suppose problem P16 is solved for each extreme point $y^i \in V$ such that $d^t y^i = 0$. Let $L = \text{Min}\{L_1, L_2, \ldots, L_s\}$. If $L \geq 0$, then $\phi(\varepsilon x, y^i) \geq \phi(0,0) \geq k$ for all $\varepsilon \geq 0$. If $L < 0$, an upper bound on ε is $d^t y^r / -L > 0$, where $d^t y^r = \text{Min}\{d^t y^i | d^t y^i > 0,\ y^i \in V\}$.

In the proof of part (ii), essentially what we have shown is that if $\phi(0,y^i) > k$ for some $y^i \in V$, then the origin of X_0 is an interior point of the closed halfspace defined by: $\phi(y^i)^+ = \{x | \phi(x,y^i) \geq k\}$. If $\phi(0,y^i) = k$, then the rays ξ^j incident on the origin are entirely contained in the closed halfspace $\phi(y^i)^+$. In both cases, we can find a point on each ray close enough to the origin and contained in $Y^0(k)$, thus implying that $Y^0(k) \cap \text{r.i.}(\xi^j) \neq \phi$.

In order to define the cutting plane, all that remains to be done is to specify values for $\bar{\lambda}_j$ as specified in Theorem 4.6. By definition,

$$\bar{\lambda}_j = \underset{\lambda>0}{\text{Max}} \; \{\underset{y\epsilon Y_0}{\text{Min}} \;\; c^t(\bar{x}-\bar{e}^j\lambda_j)+d^ty+(\bar{x}-\bar{e}^j\lambda_j)^tCy \geq k\}$$

This amounts to solving m parametric linear programming problems over Y_0. Let us now look at one way of solving a problem of this kind.

4.4 Solution of the Parametric Problem

From the definition of $\bar{\lambda}_j$ as given in Theorem 4.6, the range for each $\bar{\lambda}_j$ is $(0,\infty]$. Let $\psi(\lambda_j) = \underset{y\epsilon Y_0}{\text{Min}} \; \{c^t(\bar{x}-\bar{e}^j\lambda_j)+d^ty+(\bar{x}-\bar{e}^j\lambda_j)^tCy\}$.

Theorem 4.8. $\psi(\lambda_j)$ is a concave function of λ_j.

Proof. See [23].

Since $\psi(\lambda_j)$ is a concave function, it is unimodal. We will conduct a search over a finite range for λ_j using the ideas of the Bisecting or Bolzano Search. At each stage, we will be able to eliminate half of the current interval from consideration. This is more efficient than the method of Golden Section, in which the new interval is 0.62 times the old interval. For some $\lambda_j = \bar{\lambda}_j$, if $\psi(\bar{\lambda}_j) - k \leq \varepsilon$, for some given permissible positive tolerance level ε, we terminate the search.

If $\psi(\lambda_j) \geq k$ for all $\lambda_j > 0$, then from the statement of Theorem

4.6 we know that $\bar{\lambda}_j = \infty$. For implementation on a computer, we need to define a large number which λ_i will be set equal to instead of ∞. We will come back to some criteria for choosing this number after we present an algorithm for solving the parametric problem. The algorithm is as follows:

1. Define a large number $L >> 0$ and a permissible error $\varepsilon > 0$.
2. Solve the following linear program P17 with $\lambda_j = L$:

   P17: Minimize $c^t(\bar{x}-\bar{e}^j\lambda_j) + d^t y + (\bar{x}-\bar{e}^j\lambda_j)^t Cy \equiv \psi(\lambda_j)$

   Subject to $y \in Y_0$.

   If $\psi(L) \geq k$, terminate with $\bar{\lambda}_j = L$, otherwise go to 3.
3. Define $\lambda_H = L$, $\lambda_L = 0$, $\lambda_R = L/2$.
4. Solve P17 with $\lambda_j = \lambda_R$. If $0 \leq \psi(\lambda_R) - k \leq \varepsilon$, then terminate with $\bar{\lambda}_j = \lambda_R$. Otherwise go to 5.
5. If $\psi(\lambda_R) > k$, set $\lambda_L = \lambda_R$, $\lambda_R = (\lambda_L+\lambda_H)/2$ and go to 4; otherwise go to 6.
6. If $\psi(\lambda_R) < k$, set $\lambda_H = \lambda_R$, $\lambda_R = (\lambda_L+\lambda_H)/2$ and go to 4.

Having defined L, we have defined an interval of uncertainty for $\bar{\lambda}_j$ as (0,L) over which a binary search is conducted. Step 2 tests whether or not $\bar{\lambda}_j \geq L$. The rationale behind the stopping rule in step 4 is the fact that the exact point at which each ray intersects the polar set is really not needed. A point within the polar and close enough to the boundary would suffice. ε measures the permissible deviation from the boundary. Since the algorithm terminates in step 4, the point found is within ε of the boundary of $Y^0(k)$. In steps 5 and 6 the new interval of uncertainty is determined, which is 1/2 the previous interval. The algorithm will converge in at most (n+1) iterations, where n is the smallest positive integer for which $L/2^n \leq \varepsilon$. We can now see that we would like to make L as large as possible so as to generate a deep cut, but this may have the effect of requiring a large number of iterations before the parametric problem is solved. There are comparable interests in choosing a value for ε. So far as solving the series of problems P17 is concerned, we can always start with the previous solution, which obviously remains feasible to the next problem in the sequence, and work towards the optimal solution. Hopefully, it will be close to the optimal solution. The algorithm is very easy to implement on a computer and fairly efficient.

4.5 Statement of Cutting Plane Algorithm

We now state our Cutting Plane Algorithm for solving problem BLP 1.

1. At stage i, if X_0^i is empty, terminate; otherwise find a pseudo-global minimum $(\bar{x}^i, \bar{y}^i)$ by the algorithm of Section 3.5. Let $\bar{z} = \phi(\bar{x}^i, \bar{y}^i)$. Set $k_i = \min\{\bar{z}, k_{i-1}\}$, where k_{i-1} is the best value of the objective function ϕ at stage (i-1).

2. For each $j \in J^i$ solve the following parametric problem by the method of Section 4.4.

$$\underset{\lambda_j > 0}{\text{Max}}\ [\underset{y \in Y_0}{\text{Min}}\ c^t(\bar{x}^i - \bar{e}^j\lambda_j) + d^t y + (\bar{x}^i - \bar{e}^j\lambda_j)^t Cy \geq k_i\} = \bar{\lambda}_j$$

where $\bar{e}^j$, $j = 1,\ldots,m$ are obtained as shown in Section 3.3.

3. Define $X_0^{i+1} = X_0^i \cap \{p \mid \sum_{j\in J^i} p_j/\bar{\lambda}_j \geq 1\}$. Go to 1.

We will first compare our algorithms with some related algorithms. We will then prove infinite convergence of our algorithm, and use it to show finite convergence to an ε-optimal solution.

4.6 Comparison With Other Cutting Plane Algorithms

We would now like to compare the strength of the cuts generated by our algorithm with that proposed by Konno [26] for solving the bilinear problem. Konno first finds a local star minimum $(\bar{x},\bar{y})$. His cut is of the form $\sum_{j\in J} d_j p_j \geq \sigma$ where the d_j are positive constants which are selected and σ is a parameter which is defined to satisfy certain conditions. Let the current global minimum of $\phi(x,y)$ be k. Let C be the cone defined with vertex at $\bar{x}$ and with extreme rays corresponding to the edges incident on $\bar{x}$. Let C be bounded by the hyperplane $\sum_{j\in J} d_j p_j \leq \sigma$. Konno considers the problem:

$$\underset{\sigma>0}{\text{Max}}\ [\text{Min}\{\phi(x,y) \mid x=\bar{x}-\sum_{j\in J} \bar{e}^j p_j,\ p_j\geq 0,\ \sum_{j\in J} d_j p_j \leq \sigma,\ y\in Y_0\} \geq k]$$

He shows that for any fixed $y \in Y_0$, the minimum of ϕ will be either at $p_j = 0$ for all $j \in J$, or there will be precisely one p_j non-zero, and defined by $p_r = \sigma/d_r$. Hence, we are looking for that value of σ, say σ_r, such that $\phi(\bar{x}-\bar{e}^r \frac{\sigma_r}{d_r}, y) \geq k$ for all $y \in Y_0$. But this implies that $(\bar{x}-\bar{e}^r \frac{\sigma_r}{d_r}) \in Y^0(k)$. Now, $\bar{\lambda}_r = \max\{\lambda_r \mid (\bar{x}-\bar{e}^r\lambda_r) \in Y^0(k)\}$. Hence $\sigma_r/d_r \leq \bar{\lambda}_r$. Konno calculates σ_j/d_j for each $j \in J$ such that $(\bar{x}-\bar{e}^j \frac{\sigma_j}{d_j}) \in Y^0(k)$ and then selects $\bar{\sigma}$ such that $\bar{\sigma}/d_j \leq \sigma_j/d_j$ for all $j \in J$. Then his cut is of the form $\sum_{j\in J} d_j p_j \geq \bar{\sigma}$. But since $\sigma_j/d_j \leq \bar{\lambda}_j$, $\bar{\sigma}/d_j \leq \bar{\lambda}_j$ for all $j \in J$, that is, $d_j/\bar{\sigma} \geq 1/\bar{\lambda}_j$. This means that the cuts generated from $Y^0(k)$ are stronger than those generated by Konno's method.

There is a revealing geometric interpretation to this. Konno predetermines the coefficients d_j of the cutting plane so as to simplify computational work. But this has the effect of fixing the slope of the hyperplane. The hyperplane is now translated parallel to itself till such time as one point on it touches the boundary of $Y^0(k)$. The polar cut allows the additional flexibility of altering the slope of the cutting plane so as to cut off more of the feasible region.

We will now illustrate by means of a numerical example how our cut and that due to Konno are generated. It will also illustrate how the polar cuts are stronger than Konno's. We will use Example Problem I of Section 3.2. Let us first generate the polar cut. The pseudo-global minimum is $\bar{x} = (2,16)$ and $\bar{y} = (6,0)$ with $k = 12$. The x-tableau corresponding to this point is

x_1	x_2	u_1	u_2	u_3	u_4	b
1	0	0	-1/2	0	-1/2	2
0	1	0	0	0	1	16
0	0	1	-1/2	0	5/2	20
0	0	0	1/2	1	1/2	25.

A cut of the form $u_2/\bar{\lambda}_{u_2} + u_4/\bar{\lambda}_{u_4} \geq 1$ is desired

$$\bar{\lambda}_{u_2} = \underset{\lambda_{u_2}>0}{\text{Max}} \left\{ \underset{(y_1 y_2)\epsilon Y_0}{\text{Min}} \left[\begin{pmatrix} 2 \\ 16 \end{pmatrix} - \lambda_{u_2} \begin{pmatrix} -1/2 \\ 0 \end{pmatrix} \right]^t \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} \geq 12 \right\}$$

or

$$\underset{\lambda_{u_2}>0}{\text{Max}} \{ \underset{(y_1 y_2)\epsilon Y_0}{\text{Min}} \ (2+\tfrac{1}{2}\lambda_{u_2})y_1+16y_2 \geq 12\}$$

The solution is $\lambda_{u_2} = \infty$.

$$\bar{\lambda}_{u_4} = \underset{\lambda_{u_4}>0}{\text{Max}} \left\{ \underset{(y_1 y_2)\epsilon Y_0}{\text{Min}} \left[\begin{pmatrix} 2 \\ 16 \end{pmatrix} - \lambda_{u_4} \begin{pmatrix} -1/2 \\ 1 \end{pmatrix} \right]^t \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} \geq 12 \right\}$$

or

$$\underset{\lambda_{u_4}>0}{\text{Max}} \{ \underset{(y_1 y_2)\epsilon Y_0}{\text{Min}} \ [(2+\tfrac{1}{2}\lambda_{u_4})y_1+(16-\lambda_{u_4})y_2] \geq 12\}$$

The solution is $\lambda_{u_4} = 14.8$.

The desired cut is $u_4 \geq 14.8$ or $x_2 \leq 1.2$. Konno's cut is of the form $d_2u_2 + d_4u_4 \geq \sigma$. We select $d_2 = d_4 = 3$ according to the rules specified by Konno. σ_{u_2} is the solution to the problem:

$$\underset{\sigma_{u_2}>0}{\text{Max}} \left\{ \underset{(y_1 y_2)\epsilon Y_0}{\text{Min}} \left[\begin{pmatrix} 2 \\ 16 \end{pmatrix} - \frac{\sigma_{u_2}}{3} \begin{pmatrix} -1/2 \\ 0 \end{pmatrix} \right]^t \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} \geq 12 \right\}$$

or

$$\underset{\sigma_{u_2}>0}{\text{Max}} \left\{ \underset{(y_1 y_2)\epsilon Y_0}{\text{Min}} \left(2 + \frac{\sigma_{u_2}}{6}\right) y_1 + 16y_2 \geq 12 \right\}$$

The solution is $\sigma_{u_2} = \infty$.

σ_{u_4} is the solution to the problem:

$$\underset{\sigma_{u_4}>0}{\text{Max}} \left\{ \underset{(y_1 y_2)\epsilon Y_0}{\text{Min}} \left[\begin{pmatrix} 2 \\ 16 \end{pmatrix} - \frac{\sigma_{u_4}}{3} \begin{pmatrix} -1/2 \\ 1 \end{pmatrix} \right]^t \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} \geq 12 \right\}$$

or

$$\underset{\sigma_{u_4}>0}{\text{Max}} \left\{ \underset{(y_1 y_2)\epsilon Y_0}{\text{Min}} \left(2 + \frac{\sigma_{u_4}}{6}\right) y_1 + (16-\sigma_{u_4}/3) y_2 \geq 12 \right\}$$

The solution is $\sigma_{u_4} = 44.\dot{4}$.

$$\text{Min} \left\{ \frac{\infty}{3}, \frac{44.\dot{4}}{3} \right\} = \frac{44.4}{3}$$

Hence $\bar{\sigma} = 44.4$.

The desired cut is $3u_2 + 3u_4 \geq 44.4$ or

$$x_1 \geq 9.4.$$

Both these cuts are illustrated in the diagram shown below.

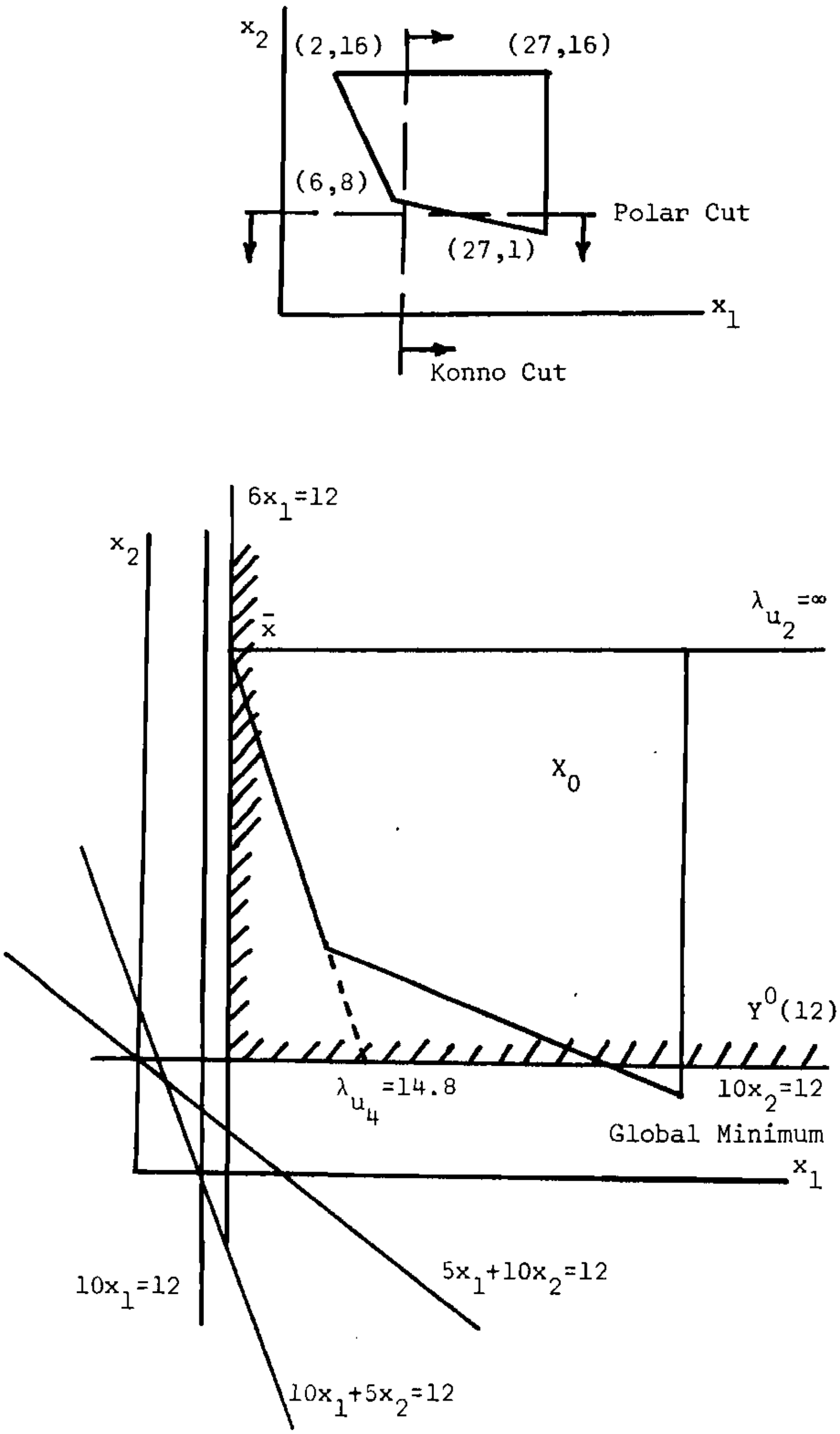


Figure 3. Example of Polar Set $Y^0(k)$

The polar set $Y^0(k)$ for this example is shown in Figure 3 with $k = 12$. $Y^0(k)$ is the set of all $x \in R^2$ satisfying

$$
\begin{aligned}
10x_2 &\geq 12 \\
10x_1 &\geq 12 \\
6x_1 &\geq 12 \\
5x_1 + 10x_2 &\geq 12 \\
10x_1 + 5x_2 &\geq 12
\end{aligned}
$$

We have shown in Section 3.2 that the Bilinear Problem can be formulated as a general nonconvex quadratic programming problem (Problem BLP 2). In Chapter I, we have reviewed a number of algorithms for solving such a problem. Any of these algorithms can be used to solve the Bilinear Problem. However, they do not take advantage of the special structure of the Bilinear Problem and will not maintain its separable structure. We cannot make a direct comparison since computational results are not available for the general algorithms.

Konno [26] has specialized Ritter's [38] algorithm for solving a general nonconvex quadratic programming problem to handle the Bilinear Problem and take advantage of its special structure. Balas and Burdet [4] used polar sets to develop a cutting plane algorithm for maximizing a convex quadratic function subject to linear constraints. They show that their cuts are deeper than those of Ritter and Tui [46] for a similar problem. We have specialized their ideas to solve the Bilinear Problem and utilize its special properties. We have shown that our cuts are deeper than those of Konno. Balas [3] has shown how to solve a non-convex quadratic problem using generalized polars based on complementary slackness conditions. But this also does not consider the special properties of the problem.

In Chapter II, we have shown that a convex quadratic maximum problem can be transformed into a bilinear problem, and hence our algorithm can be applied to solve such a problem. We have compared our algorithm with that of Moreno [33] for the convex problem. Our algorithm is 2 to 5 times faster.

4.7 Convergence of the Cutting Plane Algorithm

We will prove infinite convergence of our cutting plane algorithm with the help of the Cutting-Plane Convergence Theorem developed by Zangwill [48]. This theorem specifies a set of sufficient conditions for convergence of a cutting-plane algorithm. We will first explain these conditions and then we will show how our algorithm satisfies these conditions.

Given a set $Z^i \subset R^m$, a cutting plane algorithm determines a half-space H^i. Then the successor set $Z^{i+1} = Z^i \cap H^i$. The objective of the algorithm is to calculate a point satisfying certain properties, at which stage the algorithm will terminate. Using Z^i and a special map Γ, the set $\Gamma(Z^i)$ is defined. A test point $w^i \in \Gamma(Z^i)$ is calculated. We then verify whether or not w^i possesses the desired properties by applying a certain test, called a solution test, to it. If it passes the test, the procedure terminates. If not, we employ a second map Δ and define the set $\Delta(w^i)$. Using this map Δ, we locate a point $v^i \in \Delta(w^i)$.

Suppose we have functions a and b defined as follows: $a : R^n \to R^1$ and $b : R^n \to R^m$. The point $v^i \in R^n$ and the functions a and b define the halfspace $H_i^+(v^i) = \{x | a(v^i)+b(v^i)^t x \geq 0\}$. The halfspace $H_i^+(v^i)$ has the property that $w^i \notin H_i^+(v^i)$. The algorithm will generate the possibly infinite sequence of sets $\{z^i\}$ and points $\{w^i\}$ and $\{v^i\}$. The convergence Theorem 4.9 guarantees that the sequence $\{w^i\}$ will converge to a point w, where w passes the solution test, if a set of conditions are satisfied. We will state and prove a slightly modified version of Zangwill's Convergence Theorem, which will be applicable to our problem. Instead of using a point to set map Δ, we will consider Δ to be a function, so that $v^i = \Delta(w^i)$.

Theorem 4.9. Let a cutting-plane algorithm generate a sequence of sets $\{z^i\}$ and a corresponding sequence of points $\{w^i\}$ and $\{v^i\}$. Suppose

1. All points $\{w^i\}$ are on a compact set.
2. For any z^i, $w^i \in \Gamma(z^i)$ implies $w^i \in z^i$.
3. The function $\Delta(w^i)$ is continuous at each w^i that does not pass the solution test. Also, the functions a and b are continuous.
4. If w^i does not pass the solution test, then for $v^i = \Delta(w^i)$, $w^i \notin H_i^+(v^i) = \{x | a(v^i)+b(v^i)^t x \geq 0\}$ and $z^i \cap H_i^+(v^i) \neq \phi$.

Then if the algorithm satisfies these four conditions, the sequence $\{w^i\}$ converges to w, where w passes the solution test.

Proof. We will only consider the case for which an infinite sequence $\{z^i\}$ is generated, since, if the sequence is finite, some w^i will have passed the solution test.

From condition (4), we observe that all z^i exist and are not null. Since, from condition (1), all w^i are in a compact set, there must exist a sequence $\{w^i\}$ converging to a point w. Since the function Δ is continuous at each w^i that does not pass the solution test (condition (3)), there must exist a sequence of points $v^1, v^2, \ldots, v^i = \Delta(w^i)$, converging to v, with $v = \Delta(w)$. From condition (2), we know that $w^{i+1} \in Z^{i+1}$. Since $Z^{i+1} = Z^i \cap H_i^+(v^i)$, hence $w^{i+1} \in H_i^+(v^i)$. In general, $w^j \in H_i^+(v^i)$ for all $j \geq i + 1$, that is $a(v^i) + b(v^i)^t w^j \geq 0$ for all $j \geq i + 1$. In particular, for the limit point w, $a(v^i) + b(v^i)^t w \geq 0$. From condition (3) a and b are continuous functions of v^i and we have shown that $a(v^i) + b(v^i)^t w \geq 0$. Hence, for the limit point v, $a(v) + b(v)^t w \geq 0$, that is, $w \in H^+(v)$. Suppose w does not pass the solution test, then for $v = \Delta(w)$ we have, from condition (4) that $w \notin H^+(v)$. But we have shown that $w \in H^+(v)$, a contradiction. Hence w must pass the solution test.

We will now relate our cutting plane algorithm to the conditions specified in Theorem 4.9. We recall that the cut adjoined to the set X_0 at stage i is of the form $\sum_{j\in J} p_j^i/\bar{\lambda}_j^i \geq 1$, where p^i is a n_1 vector consisting of the x variables and the slack variables u. We will rewrite this in vector notation as $(g^i)^t p \geq 1$, where $g_j^i = 1/\bar{\lambda}_j^i$.

For simplicity, let us suppose that the cutting plane was defined with the origin transferred to the current pseudo-global minimum $(\bar{x}^i, \bar{y}^i)$. Then $\bar{\lambda}_j^i$ is the point of intersection of the cutting plane with the j*th* coordinate axis. We had determined $\bar{\lambda}_j^i$ as follows. If $\hat{k}$ was the current best value of the objective function $\phi(x,y)$, we had set

$k = \min\{\hat{k}, \phi(\bar{x}^i, \bar{y}^i)\}$ and defined the polar $Y^0(k) = \{x | \phi(x,y) \geq k$, for each $y \in Y_0\}$. Then $\bar{\lambda}_j^i$ was the intersection of the j*th* coordinate axis with the boundary of $Y^0(k)$ and hence satisfied the inequality

$$\bar{\lambda}_j^i = \underset{\lambda_j^i > 0}{\mathrm{Max}}\{\lambda_j^i | c_j \lambda_j^i + d^t y + \lambda_j^i c^j y \geq k \text{ for each } y \in Y_0\}$$

where c^j is the j*th* row of the matrix C.

We will now define the maps Δ and Γ, the functions a and b and the sequences $\{z^i\}$, $\{w^i\}$ and $\{v^i\}$ for our cutting plane algorithm. We generate a sequence of sets $\{z^i\}$ where $z^1 = (X_0 \times Y_0) \subset R^{m \times n}$, $X_0 \subset R^m$, $Y_0 \subset R^n$. Given z^i, then $z^{i+1} = z^i \cap H_i^+$, where H_i^+ is a closed halfspace and will be defined below. The map Γ is defined on the set z^i as: $\Gamma(z^i) = \{(\bar{x}^i, \bar{y}^i) | (\bar{x}^i, \bar{y}^i)$ is a pseudo-global minimum of $z^i\}$. The sequence of points $\{w^i\}$, $w^i \in \Gamma(z^i)$ constitutes the pseudo-global minimum actually located at each stage. The function $\Delta : R^{m \times n} \to R^{n_1+1}$ generates the sequence $\{v^i\}$ with $\Delta(w^i) = v^i$. If the cutting plane generated at the i*th* stage is $(g^i)^t p \geq 1$, then $v^i = (g^i, -1)^t$ defines the map Δ. We define the following functions: $a : R^{n_1+1} \to R^1$, $b : R^{n_1+1} \to R^{n_1}$, with $a(v^i) = -1$, $b(v^i) = g^i$. Then the halfspace $H_i^+(v^i) = \{p | a(v^i) + b(v^i)^t p \geq 0\}$. The algorithm terminates when the following Solution Test is passed: $z^i \cap H_i^+(v^i) = \phi$.

We will now verify that the four conditions stated in Theorem 4.9 above are satisfied.

1. All points w^i are in the compact set $(X_0 \times Y_0)$, where X_0 and Y_0 are compact by assumption.

2. For any z^i, $w^i = (\bar{x}^i,\bar{y}^i) \in \Gamma(z^i)$ implies $(\bar{x}^i,\bar{y}^i) \in z^i$, since $(\bar{x}^i,\bar{y}^i)$ is a pseudo-global minimum of z^i.

3. We want to show that the function $\Delta(w)$ is continuous for any w that does not pass the solution test. To show continuity of Δ, we need to show that if there is a sequence of points $w^1,w^2,\ldots$ converging to w, then there is a corresponding sequence of points $v^1,v^2,\ldots$ converging to v, where $\Delta(w) = v$. But each point v^i corresponds to a hyperplane which is defined by $\bar{\lambda}^i = (\bar{\lambda}_1^i,\ldots,\bar{\lambda}_{n_1}^i)$. Each $\bar{\lambda}_j^i$ is the point of intersection of the cutting plane with the j*th* coordinate axis. If $\bar{\lambda}$ is the vector corresponding to v, we will have shown that $\Delta(w)$ is continuous if we show that as $w^i \to w$, $\bar{\lambda}^i \to \bar{\lambda}$. As we have shown above, $\bar{\lambda}_j^i$ is a point on the boundary of $Y^0(k_i) = \{x|\phi(x,y)\geq k_i$, for each $y\in Y_0\}$ where $k_i = \min\{\phi(w^i),\hat{k}\}$ $\hat{k}$ being the current best solution obtained thus far. Hence $\bar{\lambda}_j^i$ solves problem P18 given below.

$$\text{P18:} \quad \text{Max}\{\lambda_j^i|c_j\lambda_j^i+d^ty+\lambda_j^ic^jy\geq k_i \text{ for each } y\in Y_0\}$$

where c^j is the j*th* row of C. Likewise, $\bar{\lambda}_j$ solves problem P19:

$$\text{P19:} \quad \text{Max}\{\lambda_j|c_j\lambda_j+d^ty+\lambda_jc^jy\geq k \text{ for each } y\in Y_0\}$$

We observe that $\bar{\lambda}_j^i$ and $\bar{\lambda}_j$ are the unique scalars solving P18 and P19, respectively. As $k_i \to k$, we see that problem P18 becomes the same as problem P19 whose solution is $\bar{\lambda}_j$. Hence, as $w^i \to w$ and consequently $k_i \to k$, $\bar{\lambda}_j^i \to \bar{\lambda}_j$ for each $j = 1,\ldots,n$. Hence, as $w^i \to w$, $\bar{\lambda}_j^i \to \bar{\lambda}_j$ which is the required result.

We have defined $b(v^i) = g^i$ and $a(v^i) = -1$, where $v^i = (g^i,-1)^t$ which are obviously continuous functions.

4. If $z^i \cap H_i^+(v^i) \neq \phi$, w^i does not pass the solution test. Since the cutting plane was defined with the origin transferred to w^i, we have $w^i = 0$. Hence, $a(v^i) + b(v^i)^t p = a(v^i) = -1 < 0$. Hence $w^i \notin H_i^+(v^i) = \{p|a(v^i)+b(v^i)^t p \geq 0\}$.

We have thus shown that all the four conditions of Theorem 4.9 are satisfied. Hence our cutting plane algorithm will converge to the global minimum.

4.8 Finite Algorithm for an ε-Optimal Solution

We have shown above that the algorithm converges to an optimal solution at least infinitely. We will show below how a near-optimal solution can be obtained in a finite number of steps. Moreover, in this case, it will be adequate to work with a local star minimum at each stage rather than a pseudo-global minimum. Hence, step 3 of Algorithm A in Chapter III (for finding a pseudo-global minimum) can be deleted.

Definition 4.10. $(\bar{x},\bar{y})$ is an *ε-optimal solution* to BLP 1 if, given an $\varepsilon > 0$, $\bar{x} \in X_0$, $\bar{y} \in Y_0$, then $\phi(\bar{x},\bar{y}) \leq \phi(x,y) + \varepsilon$ for all $x \in X_0$ and $y \in Y_0$.

The strategy used in getting an ε-optimal solution to BLP 1 is to partition X_0 into a finite number of regions such that within each region we know the ε-optimal solution for all $y \in Y_0$. For this purpose we will use the polar set $Y^0(k-\varepsilon)$ at each stage rather than $Y^0(k)$. We will show that a valid cut can be generated from a local star minimum $(\bar{x},\bar{y})$ by using $Y^0(k-\varepsilon)$, where k is the current best value of $\phi(x,y)$, which may or may not be attained at $(\bar{x},\bar{y})$. Here, by a valid cut we mean that the cut deletes the point $(\bar{x},\bar{y})$, but not any feasible point (x,y) such that $\phi(x,y) < k - \varepsilon$, for some $y \in Y_0$. Lemma 4.11 below is a modified version of Lemma 4.7 and gives the properties of:

$$Y^0(k-\varepsilon) = \{x \mid \phi(x,y) \geq k-\varepsilon \text{ for each } y \in Y_0\}$$

Lemma 4.11. Let $(\bar{x},\bar{y})$ be a local star minimum of BLP 1. For a given $\varepsilon > 0$, $Y^0(k-\varepsilon)$ is a closed convex set having the properties:

(i) $\bar{x} \in Y^0(k-\varepsilon)$.

(ii) $Y^0(k-\varepsilon) \cap \text{r.i.}(\xi^j) \neq \phi$ for each $j \in J$.

(iii) $Y^0(k-\varepsilon) \cap \{x \in X_0 \mid \phi(x,y) < k-\varepsilon \text{ for some } y \in Y_0\} = \phi$.

Proof. From Lemma 4.2, $Y^0(k-\varepsilon)$ is clearly a closed convex set. Also $\phi(\bar{x},y) \geq \phi(\bar{x},\bar{y}) \geq k > k - \varepsilon$ for each $y \in Y_0$ since $(\bar{x},\bar{y})$ is a local star minimum. Hence $\bar{x} \in \text{int } Y^0(k-\varepsilon)$. Hence (i) and (ii) hold. Part (iii) is satisfied from the definition of $Y^0(k-\varepsilon)$.

Thus $Q = Y^0(k-\varepsilon)$ satisfies the hypothesis of Theorem 4.6 (with k replaced by k-ε) and $\sum_{j \in J} p_j/\bar{\lambda}_j \geq 1$ is a valid cutting plane in the sense that it does not cut off any feasible point x with $\phi(x,y) < k - \varepsilon$ for some $y \in Y_0$. Hence k is the global ε-minimum value over the X_0 region already explored (including the region cut off by the cutting plane) and all the Y_0 region.

Note that only the local star minimum property of $(\bar{x},\bar{y})$ has been invoked. In Lemma 4.7, we needed $(\bar{x},\bar{y})$ to be a pseudo-global minimum only to prove part (ii) of the lemma.

We can now summarize the ε-optimal algorithm as follows:

Step 1. At stage i, if X_0^i is empty, terminate. Otherwise find the local star minimum $(\bar{x}^i,\bar{y}^i)$ using Steps 1 and 2 of Algorithm A (Section 3.5). Let $\bar{Z} = \phi(\bar{x}^i,\bar{y}^i)$. Set $k = \min\{\bar{z},k_{i-1}\} - \varepsilon$, where k_{i-1} is the best value of the objective function ϕ at stage i - 1.

Steps 2 and 3. As in Steps 2 and 3 of the cutting plane algorithm of Section 4.5.

We will now show that the algorithm is finite. At stage i, suppose the local star minimum is $(\bar{x},\bar{y})$ and let k_i given at Step 1 of the algorithm correspond to $(\hat{x},\hat{y})$, which may or may not be the same as $(\bar{x},\bar{y})$. Then $(\hat{x},\hat{y})$ is the ε-optimal solution over all $y \in Y_0$ and over X_0 already explored, i.e., for all x in X_0, but not in X_0^{i+1}. Note that k_i is decreased at each stage at least by a fixed $\varepsilon > 0$. Hence, if we show that for the global optimum $(\bar{\bar{x}},\bar{\bar{y}})$ of BLP 1, $\bar{\bar{x}}$ is cut off from a cut obtained from $Y^0(\bar{\bar{k}})$ for some $\bar{\bar{k}}$, then the algorithm is finite.

Now consider BLP 3 with $(\bar{x},\bar{y})$ as the local star minimum. If $(\bar{\bar{x}},\bar{\bar{y}}) \in X_0^{i+1}$ is the global minimum, then $\bar{\bar{x}}$ is in the cone with vertex $\bar{x}$ and ξ^j, $j \in J$, as the generators. Then $\bar{\bar{x}}$ can be expressed as a convex combination of points $\bar{x}^j$, $j \in J$, on these generators.

Let $\alpha_j = \min_{y \in Y_0} \phi(\bar{x}^j,y)$ and $\bar{\bar{k}} = \min \alpha_j$. Then $Y^0(\bar{\bar{k}})$ will cut the rays

ξ^j at points $(0,\ldots,\bar{\lambda}_j,\ldots,0) \geq \bar{x}^j$. Hence $\sum_{j\in J} p_j/\bar{\lambda}_j \geq 1$ obtained from $Y^0(\bar{k})$ will cut off $\bar{\bar{x}}$. This shows that the ε-optimal algorithm is finite.

CHAPTER V

ALGORITHM BASED ON INDUCTIVE CONSTRUCTION OF POLYTOPES

5.1 Introduction

In this chapter, we will be solving problem BLP 1 by constructing a sequence of polytopes. We restate problem BLP 1:

$$\text{BLP 1} \quad \text{Minimize} \quad \phi(x,y) = c^t x + d^t y + x^t C y$$

$$\text{Subject to} \quad x \in X_0 = \{x \mid Ex \le e,\ x \ge 0\}$$

$$y \in Y_0 = \{y \mid Fy \le f,\ y \ge 0\}$$

where X_0 and Y_0 are bounded polytopes.

The basic ideas of the method for locating the global minimum of $\phi(x,y)$ to be developed in this chapter are as follows. Let P be a polytope in R^m such that there is at least one extreme point $\hat{x}$ of X_0 which is also an extreme point of P. Suppose we know the equations of the hyperplanes which are (m-1) dimensional faces of P. Suppose also that we know the global minimum of ϕ over P for all $y \in Y_0$ is $\hat{x}$. If $X_0 \subset P$, then clearly we have found the global minimum of ϕ over X_0 and Y_0. If not, we will find an extreme point x^k of X_0 which is not in P, and define the (m-1) dimensional faces of a new and enlarged polytope P' which contains P and x^k. We will now find $\min_{y \in Y_0} \phi(y,x^k)$. P' is defined such that we now have the global minimum of ϕ over P' and Y_0 known. After a finite number of steps, the polytope P will have been "enlarged" sufficiently such that the condition $X_0 \subset P$ holds, at which point the procedure terminates.

5.2 Preliminary Properties

Definition 5.1. Let P be a convex subset of R^m. A set F, $F \subset P$ is a *face* of P if either $F = \phi$, or $F = P$, or if there exists a supporting hyperplane H of P such that $F = P \cap H$. ϕ and P are called the *improper* faces of P. All other faces are called *proper* faces. Note that a face is a convex set.

Definition 5.2. A maximal proper face of P, that is, a proper face of the highest possible dimension, is called a *facet* of P. Thus if P is an m-dimensional polytope, a facet of P is (m-1)-dimensional.

Definition 5.3. Given a set A, the *affine hull* of A is the set aff $[A] = \{x | x = \sum_{i=1}^{k} \lambda_i x^i, \sum_{i=1}^{k} \lambda_i = 1, x^i \epsilon A, i=1,\ldots,k\}$ for arbitrary finite k.

Definition 5.4. Given a set A, the *convex hull* of A is the set $\text{conv}[A] = \{x | x = \sum_{i=1}^{k} \lambda_i x^i, \lambda_i \geq 0, \sum_{i=1}^{k} \lambda_i = 1, x^i \epsilon A, i=1,\ldots,k\}$ for arbitrary finite k.

Definition 5.5. Let P be an m-dimensional polytope, H a hyperplane such that $H \cap \text{int}(P) = \phi$, and let $v \in R^m$. v is said to be *beneath* H (with respect to P) provided v belongs to the open halfspace determined by H which contains int(P). If F is a facet of P, v is beneath F if v is beneath aff[F].

Definition 5.6. With respect to Definition 5.5, if v is in the open halfspace determined by H which does not contain int(P), then v is said to be *beyond* H. Similarly, v is beyond F if v is beyond aff[F].

The concepts of beneath and beyond will be used very heavily in the inductive construction of the faces of the polytope P' from P.

Lemma 5.7. Let P and P' be two polytopes, $P \subset P'$. Let F' be a face of P'. Then $F = F' \cap P$ is a face of P.

Proof. See [21].

Suppose we are given m linearly independent points in R^m: $x^1, x^2, \ldots, x^m$. We can uniquely define a hyperplane (manifold) H passing through the m points. Let H be represented by the equation $p^t x = 1$. The hyperplane H defines two halfspaces H^+ and H^-, where

$$H^+ = \{x | p^t x \geq 1\} \quad \text{and}$$

$$H^- = \{x | p^t x \leq 1\}$$

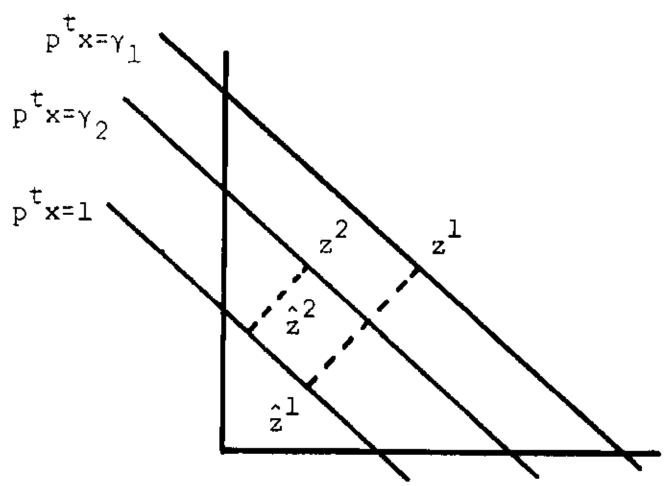
Since any $x \in R^m$ can be expressed as a linear combination of the m independent points $x^1, \ldots, x^m$, we have for any $x \in H^+$, $1 \leq p^t x = \sum_{i=1}^{m} \lambda_i p^t x^i = \sum_{i=1}^{m} \lambda_i$, that is, $\sum_{i=1}^{m} \lambda_i \geq 1$. Likewise, for $x \in H^-$, we have $\sum_{i=1}^{m} \lambda_i \leq 1$. Of course for $x \in H$, we have $\sum_{i=1}^{m} \lambda_i = 1$.

As in the notation H^+ and H^-, we will denote the two closed halfspaces defined by aff[F] of a facet by $\text{aff}[F]^+$ and $\text{aff}[F]^-$.

Lemma 5.8. Let $P \subset R^m$ be a polytope, $F = \text{conv}[x^1,\ldots,x^m]$ a facet of P, $\text{aff}[F] = \{x \in R^m | p^t x = 1\}$ and $P \subset \text{aff}[F]^- = \{x \in R^m | p^t x \leq 1\}$. Let $v \in R^m$, $v = \sum_{i=1}^m \lambda_i x^i$. v is beyond aff[F] if and only if $\sum_{i=1}^m \lambda_i > 1$. v is beneath aff[F] if and only if $\sum_{i=1}^m \lambda_i < 1$.

Proof. Let v be beyond aff[F]. Then $v \in \text{int}(\text{aff}[F]^+)$. Hence $1 < p^t v = \sum_{i=1}^m \lambda_i p^t x^i = \sum_{i=1}^m \lambda_i$. Conversely, let $\sum_{i=1}^m \lambda_i > 1$. Then $p^t v = \sum_{i=1}^m p^t \lambda_i x^i = \sum_{i=1}^m \lambda_i p^t x^i = \sum_{i=1}^m \lambda_i$ since $p^t x^i = 1$ $(i=1,\ldots,m)$. But $\sum_{i=1}^m \lambda_i > 1$. Hence $p^t v > 1$, which implies $v \in \text{int}(\text{aff}[F]^+)$. Since $P \subset \text{aff}[F]^-$, v is beyond aff[F]. The proof of the other part is identical. We note that if $P \subset \text{aff}[F]^+$, and v is beyond aff[F], then $\sum_{i=1}^m \lambda_i < 1$. If v is beneath aff[F], then $\sum_{i=1}^m \lambda_i > 1$.

Now suppose that the origin $0 \in \text{int}(H^-)$. Let us consider two points $z^1, z^2 \in \text{int}(H^+)$. Let $\hat{z}^1$ and $\hat{z}^2$ be the projections of z^1 and z^2, respectively on H.



Let $z^1 = \sum_{i=1}^{m} \alpha_i x^i$ and $z^2 = \sum_{i=1}^{m} \beta_i x^i$. Let $p^t x = \gamma_1$ and $p^t x = \gamma_2$ be the equations of the hyperplanes parallel to $p^t x = 1$ and passing through z^1 and z^2, respectively. Then $\|z^1-\hat{z}^1\| = (\gamma_1-1)/\|p\|$ and $\|z^2-\hat{z}^2\| = (\gamma_2-1)/\|p\|$. If the Euclidean distance of z^1 from H is greater than that of z^2, then $\gamma_1 > \gamma_2$. Hence $p^t z^1 = \sum_{i=1}^{m} p^t \alpha_i x^i > p^t z^2 = \sum_{i=1}^{m} \beta_i p^t x^i$. Since $p^t x^i = 1$, $i = 1,\ldots,m$, $\sum_{i=1}^{m} \alpha_i > \sum_{i=1}^{m} \beta_i$. We will be using this property in the following way.

Suppose we have a polytope $X_0 \subset R^m$ and the origin is a vertex of X_0. For $x \in X_0$, we have $x = \sum_{i=1}^{m} \lambda_i x^i$ where $\{x^i, i=1,\ldots,m\}$ are linearly independent points in R^m. We can write this as $x = B\lambda$, where $B = [x^1,x^2,\ldots,x^m]$ and $\lambda^t = (\lambda_1,\lambda_2,\ldots,\lambda_m)$. Hence $\lambda = B^{-1}x$. Let H be the hyperplane passing through the points $x^1,x^2,\ldots,x^m$. Suppose we want to find the extreme point of X_0 which is furthest from H. We can do this by solving problem P20:

$$\text{P20:} \quad \text{Maximize} \quad e^t B^{-1} x$$
$$\text{Subject to} \quad x \in X_0$$

where e is a vector of ones. Since P20 is a linear program, its optimal solution will be at an extreme point of X_0. We observe that the objective function of P20 is simply $e^t\lambda = \sum_{i=1}^{m} \lambda_i$. Let $\bar{w}$ be the maximum objective function value of P20. If $\bar{w} < 1$, we can conclude that there does not exist any point of X_0 on or beyond H.

We now address ourselves to the question of how to determine all the facets of the polytope P' assuming that all the facets of a polytope

P are known, where $P = \text{conv}[x^1,\ldots,x^m]$ and $P' = \text{conv}[x^1,\ldots,x^m,x^{m+1}]$, $x^{m+1} \notin P$. The reason why we would be interested in this is as follows. Let $\theta = \{\theta_1,\ldots,\theta_s\}$ represent the set of all facets of P. Suppose we know the global minimum of $\phi(x,y)$ for all $x \in P$, $y \in Y_0$, which is attained at $(\bar{x},\bar{y})$, where $\bar{x}$ is an extreme point of P and of X_0 and $\bar{y}$ is an extreme point of Y_0. If $X_0 \subset P$ then the problem BLP 1 is solved (see Figure 4).

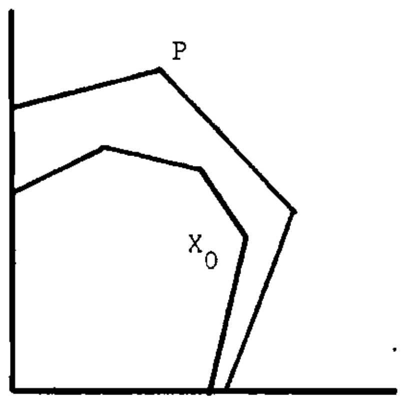


Figure 4. Example Where $X_0 \subset P$

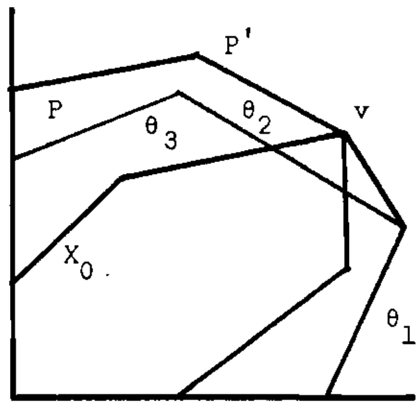


Figure 5. Example Where $X_0 \not\subset P$

If not, as in Figure 5, we will find an extreme point v of X_0, $v \notin P$. P will be suitably enlarged to P' and $v \in P'$.

To determine whether or not $X_0 \subset P$, we will solve Problem P20 for each $\theta_i \in \theta$. If a point v beyond $\text{aff}[\theta_i]$ is found, $X_0 \not\subset P$ and P is enlarged. If $v \in \text{aff}[\theta_i]$, we will check all alternative solutions to Problem P20. If all solutions yield points in P (for each θ_i), then $X_0 \subset P$. If not, we have a point $v \notin P$, and P is again enlarged. We will show later that the procedure will lead to an optimum solution in a finite number of steps.

5.3 Construction of the New Polytope P'

We will now show how a new polytope $P' \supset P$ can be defined given all the facets of a polytope P if we locate an extreme point x^k of X_0 on or beyond one of the facets of P, $x^k \notin P$. We will define P' by defining all its facets. For this purpose, we need the following theorems.

Theorem 5.9. Each m-dimensional polytope P is the intersection of a finite family of closed halfspaces; the smallest such family consists of those closed halfspaces containing P whose boundaries are the affine hulls of the facets of P.

Proof. See [21].

Theorem 5.10. If F_1 is a face of the polytope P, and if F_2 is a face of the polytope F_1, then F_2 is a face of P.

Proof. See [21].

Theorem 5.11. If P is an m-dimensional polytope, each (m-2)-dimensional face F of P is contained in precisely two facets F_1 and F_2 of P and $F = F_1 \cap F_2$.

Proof. See [21].

Theorem 5.12. Let P and P' be two m-dimensional polytopes in R^m, and let v be a vertex of P', $v \notin P$, such that $P' = \text{conv}[\{v\} \cup P]$. Then

i. A face F of P is a face of P' if and only if there exists a facet F' of P such that $F \subset F'$ and v is beneath F'.

ii. If F is a face of P, then $F' = \text{conv}[\{v\} \cup F]$ is a face of P' if and only if either (a) $v \in \text{aff}[F]$ or (b) among the facets of P containing F there is at least one such that v is beneath it and at least one such that v is beyond it.

Moreover, each face of P' is of one and only one of those types.

Proof. See [21].

<u>Theorem 5.13</u>. Let P and P' be two m-dimensional polytopes in R^m, let v be a vertex of P', $v \notin P$, such that $P' = \text{conv}[\{v\} \cup P]$. Then

i. A facet F of P is a facet of P' if and only if v is beneath aff[F].

ii. Let F be a facet of P. Then $F' = \text{conv}[\{v\} \cup F]$ is a facet of P' if and only if $v \in \text{aff}[F]$.

iii. Let F be a (m-2)-dimensional face of P. Let F_1 and F_2 be the two facets of P containing F. Then $F' = \text{conv}[\{v\} \cup F]$ is a facet of P' if and only if v is beneath aff[F_1] (or aff[F_2]) and beyond aff[F_2] (or aff[F_1]).

iv. Each facet of P' is either a facet of P or is of the form $F' = \text{conv}[\{v\} \cup F]$, where F is a face of P.

Proof.

i. Let F be a facet of P and let v be beneath F. Then from part (i) of Theorem 5.12, F is a face of P'. But F is (m-1)-dimensional since it is a facet of P. Hence it is a facet of P'.

Conversely, let F be a facet of both P and P'. Then from part

(i) of Theorem 5.12 there exists a facet $\bar{F}$ of P such that F is contained in $\bar{F}$ and v is beneath $\bar{F}$. But F and $\bar{F}$ being facets of P' and $F \subset \bar{F}$ implies $F = \bar{F}$. Hence v is beneath F.

ii. Let F be a facet of P and $v \in \text{aff}[F]$. From part (ii) (a) of Theorem 5.12, $F' = \text{conv}[\{v\} \cup F]$ is a face of P'. But F is (m-1)-dimensional, hence F' is (m-1)-dimensional and hence F' is a facet of P'.

Conversely, let F be a facet of P and let $F' = \text{conv}[\{v\} \cup F]$ be a facet of P'. Hence both F and F' are (m-1)-dimensional. Hence $v \in \text{aff}[F]$.

iii. Let F be a (m-2)-dimensional face of P, and let v be beneath F_1 and beyond F_2. From part (ii) (b) of Theorem 5.12, $F' = \text{conv}[\{v\} \cup F]$ is a face of P'. Now $v \notin \text{aff}[F_1]$ and $v \notin \text{aff}[F_2]$ so that $v \notin \text{aff}[F_1] \cap \text{aff}[F_2]$. From Theorem 5.11, $F = F_1 \cap F_2$, and hence $\text{aff}[F] = \text{aff}[F_1] \cap \text{aff}[F_2]$. Hence $v \notin \text{aff}[F]$. Hence F' is (m-1)-dimensional.

Conversely, let F be a (m-2)-dimensional face of P and F' be a facet of P'. Since F' is (m-1)-dimensional, $v \notin \text{aff}[F]$. Also from Theorem 5.11, there are precisely 2 facets F_1 and F_2 of P such that $F_1 \cap F_2 = F$. Hence from part (ii) (b) of Theorem 5.12, v is beneath F_1 (or F_2) and beyond F_2 (or F_1).

iv. From Theorem 5.12, each facet of P' is either a face of P or is of the form $\text{conv}[\{v\} \cup F]$, where F is a face of P. In the former case, F is a facet of P. In the latter case, F is (m-1)-dimensional and

$v \in$ aff[F], or F is (m-2)-dimensional and $v \notin$ aff[F].

We will now illustrate each one of these cases. In Figure 6 F_1 is a facet of P. Since v is beneath aff[F_1], F_1 is a facet of P'. In Figure 7, $v \in$ aff[F_1], and hence conv[{v}$\cup F_1$] is a facet of P'. In Figure 6, F is a (2-2)=0-dimensional face of P. There are precisely two facets F_1 and F_2 which contain F, v is beneath aff[F_1] and beyond aff[F_2]. Hence conv[F$\cup${v}] is a facet of P'.

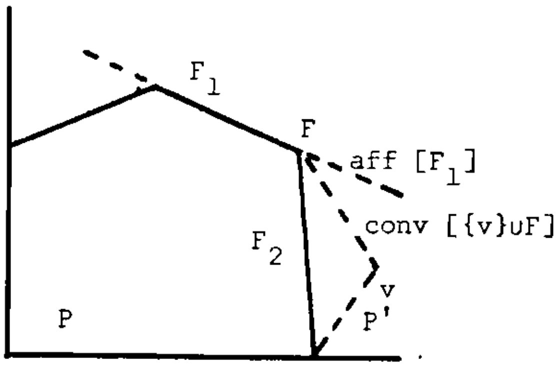


Figure 6. Examples of Facets of P'

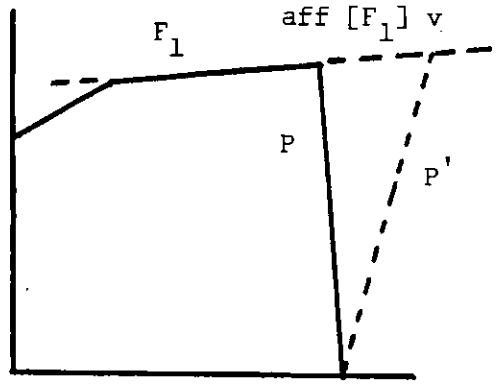


Figure 7. Examples of Facets of P'

Theorem 5.13 will enable us to define a sequence of polytopes $P_1, P_2, \ldots$ such that $P_{i+1} \supset P_i \supset \ldots$. Polytope P_i will have at least one extreme point which is also an extreme point of X_0. Consider the extreme points of P_{i+1}. This will have precisely one extreme point x^k not contained in P_i, with x^k an extreme point of X_0. At each stage we will maintain a list of all the facets of the current polytope P_i. When we find an extreme point x^k of X_0 not contained in P_i, we will generate all the facets of P_{i+1} with the help of Theorem 5.13. For any polytope P_i, we will record its *jth* facet $\theta_{ij} = \text{conv}[x^{1,i,j}, x^{2,i,j}, \ldots, x^{m,i,j}]$.

We will refer to these points as elements of a facet. We will now discuss in some detail how the facets of P_{i+1} are generated, given $x^k \notin P$ but an extreme point of X_0.

Suppose we know all the facets of a polytope P_i. Let these be $\theta_{i1}, \theta_{i2}, \ldots, \theta_{in}$. If $\text{aff}[\theta_{ij}] = \{x \in R^m | p^t x = 1\}$, we will define each facet θ_{ij} such that $P_i \subset \text{aff}[\theta_{ij}]^- = \{x \in R^m | p^t x \leq 1\}$. Let x^k be an extreme point of X_0, $x^k \notin P_i$. From Theorem 5.13, we see that in order to construct all the facets of $P_{i+1} = \text{conv}[P_i \cup \{x^k\}]$, we need to sort the facets of P_i into three classes as follows:

a. This will consist of all facets θ_{ie} of P_i such that x^k is beneath $\text{aff}[\theta_{ie}]$. From Theorem 5.13, part (i), θ_{ie} will be a facet of P_{i+1}. If $\theta_{ie} = \text{conv}[x^{1,i,e}, \ldots, x^{n,i,e}]$ and $x^k = \sum_{j=1}^{n} \lambda_j x^{j,i,e}$, then from Lemma 5.8, θ_{ie} will be in this class if $\sum_{j=1}^{n} \lambda_j < 1$.

b. This will consist of all those facets θ_{im} of P_i such that $x^k \in \text{aff}[\theta_{im}]$. From Theorem 5.13, part (ii), $\text{conv}[\theta_{im} \cup \{x^k\}]$ will be a facet of P_{i+1}. θ_{im} will be in this class if $\sum_{j=1}^{n} \lambda_j = 1$.

c. This will consist of all those facets θ_{iq} of P_i such that x^k is beyond $\text{aff}[\theta_{iq}]$. From Lemma 5.8, θ_{iq} will be in this class if $\sum_{j=1}^{n} \lambda_j > 1$. In order to use Theorem 5.13, part (iii), we will identify all (m-2)-dimensional faces F of P_i such that there exists a facet θ_{ie} of P_i in class (a) which contains F and a facet θ_{iq} of P_i in class (c) which contains F. We can do this by determining whether or not θ_{ie} and θ_{iq} have (m-1) elements in common. If they do, let these elements be $x^1, \ldots, x^{m-1}$. Then $\text{conv}[x^1, \ldots, x^{m-1}]$ is a (m-2) dimensional face of P_i which satisfies the hypotheses of Theorem 5.13, part (iii). Hence

$\text{conv}[x^1,\ldots,x^{m-1},x^k]$ is a facet of P_{i+1}.

From Theorem 5.13, part (iv), the facets generated in this manner will be all the facets of P_{i+1}.

We will now show how convergence can be improved by considering as large a polytope as possible. We will also discuss how the first polytope can be defined.

5.4 Improving Convergence

Since our procedure is to get a sequence of polytopes $P_1 \subset P_2 \subset \ldots P_n$ and our stopping rule is to have $X_0 \subset P_n$, it is obvious that convergence will be accelerated if as large a polytope as possible is generated at each stage. We will do this by the use of polar sets $Y^0(k)$ introduced in Chapter IV at every stage.

We will now show how the first polytope is defined. Let $(\bar{x},\bar{y})$ be the first pseudo-global minimum obtained as shown in Chapter III. Let us transform the origin of the coordinate system to $(\bar{x},\bar{y})$, so that the problem is of the form BLP 3. Let us restate the set X_0:

$$X_0 = \{x \in R^m | Ex \le e,\ x \ge 0\}$$

We note that $e \ge 0$, and at the origin the slack variables u are basic. Let k be the objective function value at $(\bar{x},\bar{y})$. We recall the definition of the polar set $Y^0(k)$:

$$Y^0(k) = \{x \in R^m | \phi(x,y) \ge k \text{ for each } y \in Y_0\}$$

We will consider the set X_0' which is obtained from X_0 by removing those constraints for which the current right-hand side term (e_i) is zero. Thus, $X_0 \subset X_0'$. From Theorem 3.20, the origin is a vertex of X_0' and there are m rays incident on it, the j*th* ray being $\xi^j = \{x \in R^m | x_i = 0, i \neq j, x_j \geq 0\}$. We will find the intersection of these m rays with $Y^0(k)$. In order to do so, we will solve the following parametric problem:

$$\underset{x_j > 0}{\text{Max}} \{\text{Min } c_j x_j + d^t y + x_j (c^j) y \geq k \text{ for each } y \in Y_0\}$$

where c^j is the j*th* row of C. Suppose the points of intersection of the m rays ξ^j are $\bar{x}^j$, $j = 1,\ldots,m$, where $\bar{x}^j = (0,\ldots,0,\bar{x}_j,0,\ldots,0)$. Then the initial polytope $P_1 = \text{conv}[\bar{x}^1,\ldots,\bar{x}^m,0]$. The facets of P_1 are $\text{conv}[\bar{x}^1,\ldots,\bar{x}^m]$ and $\text{conv}[\bar{x}^1,\ldots,\bar{x}^{j-1},0,\bar{x}^{j+1},\ldots,\bar{x}^m]$, $j = 1,\ldots,m$.

Certain facets of P_1 are the facets of all subsequent polytopes. Consider the hyperplane $\hat{H}_j = \text{aff}\left[\text{conv}[\bar{x}^1,\ldots,\bar{x}^{j-1},0,\bar{x}^{j+1},\ldots,\bar{x}^m]\right]$. This hyperplane can also be represented by the equation $x_j = 0$, that is, it is one of the hyperplanes defining the system of coordinates. Thus each $x \in X_0$ is beneath $\text{aff}\left[\text{conv}[\bar{x}^1,\ldots,\bar{x}^{j-1},0,\bar{x}^{j+1},\ldots,\bar{x}^m]\right]$ for $j=1,\ldots,m$. In particular every extreme point of X_0 is beneath $\text{aff}\left[\text{conv}[\bar{x}^1,\ldots,\bar{x}^{j-1},0,\bar{x}^{j+1},\ldots,\bar{x}^m]\right]$. Thus we know that $\text{conv}[\bar{x}^1,\ldots,\bar{x}^{j-1},0,\bar{x}^{j+1},\ldots,\bar{x}^m]$ will be a facet of each polytope of the sequence. Moreover, since there does not exist any extreme point of X_0 beyond $\text{aff}\left[\text{conv}[\bar{x}^1,\ldots,\bar{x}^{j-1},0,\bar{x}^{j+1},\ldots,\bar{x}^m]\right]$, we will never use these facets to find an extreme point of X_0 which is not contained in the current polytope P_i. We, therefore, use the special notation $\hat{H}_j$ for these hyperplanes.

We will next discuss how to enlarge a given polytope P_i. Let us recall that P_i is expressed as the convex hull of the points $0,\bar{x}^1,\ldots,\bar{x}^p$. Suppose each $\bar{x}^j$ has the property that $\phi(\bar{x}^j,y) \geq k_i$ for all $y \in Y_0$, where k_i is the current global minimum, i.e., each $\bar{x}^j \in Y^0(k_i)$. Suppose we locate an extreme point x^r of X_0, $x^r \notin P_i$. If $x^r \in \mathrm{int}\left(Y^0(k_i)\right)$, the point $\bar{x}^r$ where the ray $\lambda x^r, \lambda \geq 0$ intersects the boundary of $Y^0(k_i)$ is used in defining P_{i+1}. If x^r is on the boundary of $Y^0(k_i)$ the polytope is not enlarged. If $x^r \notin Y^0(k_i)$, then $\phi(x^r,y^r) = k_{i+1} < k_i$ for some $y^r \in Y_0$. Consider the polar $Y^0(k_{i+1}) = \{x|\phi(x,y)\geq k_{i+1} \text{ for all } y\in Y_0\}$. Since $k_{i+1} < k_i$ by definition of the polar sets $Y^0(k)$, we have $Y^0(k_i) \subset Y^0(k_{i+1})$. Suppose we find the intersection of each ray emanating from the origin and passing through the points $\bar{x}^1,\ldots,\bar{x}^p$. $\bar{x}^j$ was the point of intersection of the j*th* ray with the polar $Y^0(k_i)$ and since $Y^0(k_i) \subset Y^0(k_{i+1})$, the point of intersection $\hat{x}^j$ with the polar $Y^0(k_{i+1})$ will be such that $\hat{x}^j \geq \bar{x}^j$. We have defined a larger polytope such that the minimum of ϕ over this polytope for all $y \in Y_0$ is k_{i+1} and this occurs at an extreme point of X_0. Such an enlargement of the polytope will lead to a quicker attainment of the sufficient condition for termination that is $X_0 \subset P_n$, for some polytope P_n in the sequence. This is illustrated in Figure 8.

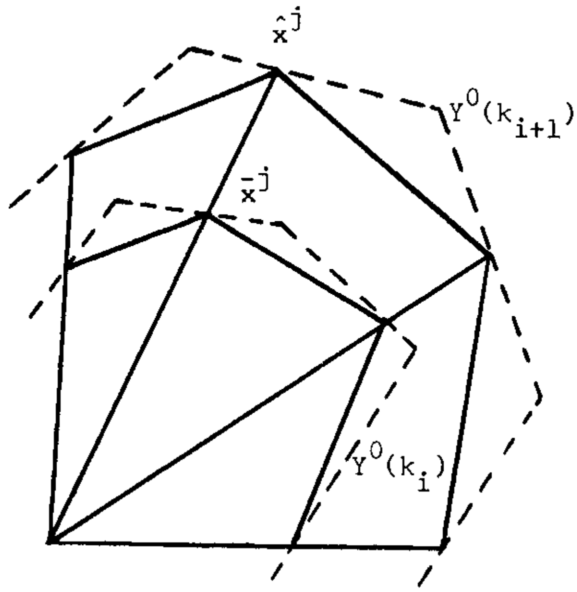


Figure 8. Enlargement of A Polytope

5.5 Algorithm for Inductive Construction of Polytopes

In this section, we will state a "schema" for constructing the sequence of polytopes.

For ease of presentation, we will define the following problems which need to be solved at different stages of the algorithm:

$$\text{I. LP1:} \quad \max_{\lambda>0} \{\min_{y\in Y_0} \lambda c^t\hat{x}+d^ty+\lambda\hat{x}^tCy\geq k\}$$

with $\hat{x}$ fixed.

$$\text{II. LP2:} \quad \text{Max} \quad z = eB_j^{-1}x$$

Subject to $x \in X_0$

with $B_j = [\bar{x}^{j_1},\bar{x}^{j_2},\ldots,\bar{x}^{j_m}]$, a given matrix and $e = [1,1,\ldots,1]$.

III. LP3: Min $z(y) = c^t\hat{x} + d^t y + \hat{x}^t Cy$

$$y \in Y_0$$

with $\hat{x}$ fixed.

(0) Find the pseudo-global minimum $(\bar{x},\bar{y})$ by Algorithm A of Section 3.5. Let $k_0 = c^t\bar{x} + d^t\bar{y} + \bar{x}^t C\bar{y}$.

(1) Transform the origin of the coordinate system to $(\bar{x},\bar{y})$ as shown in Section 3.3.

(2) Find the points $\bar{x}^1,\ldots,\bar{x}^m$ along the m rays $\xi^j = \{x \in R^m \mid x_i = 0,\ i \neq j,\ x_j \geq 0\}$ incident on 0 where the rays ξ^j intersect $Y^0(k_0)$ by solving LP1. Define $S_0 = \{\theta_0\}$, where $\theta_0 = \text{aff}\left[\text{Conv}[\bar{x}^1,\ldots,\bar{x}^m]\right]$, $\hat{H}_j = \text{aff}\left[\text{Conv}[\bar{x}^1,\bar{x}^2,\ldots,\bar{x}^{j-1},0,\bar{x}^{j+1},\ldots,\bar{x}^m]\right]$ for $j=1,\ldots,m$; $P_0 = \text{Conv}[0,\bar{x}^1,\ldots,\bar{x}^m]$.

(3) At the *ith* stage, if $S_i = \phi$, terminate. Otherwise *remove* an element $\theta_j = \text{aff}\left[\text{Conv}[\bar{x}^{j_1},\bar{x}^{j_2},\ldots,\bar{x}^{j_m}]\right]$ from S_i. Solve problem LP2 with B_j formed by the m elements of θ_j. Let $x^q \in X_0$ be a solution of LP2.

(4) If the optimal value of LP2 is $\bar{z} < 1$, go to (3). If $\bar{z} = 1$, find all alternative optimal solutions to LP2. If there exists a new solution (i.e., a point not in P_i) x^q go to (5). Otherwise go to (3). If $\bar{z} > 1$, then the solution x^q found in step (3) is an extreme point not contained in P_i, go to (5).

(5) Solve LP3 with $\hat{x} = x^q$. If the optimal value $\bar{z}(y)$ is $k_{i+1} < k_i$, set $\bar{x}^q = x^q$, and go to (6). If $\bar{z}(y) = k_i$, go to (7). Otherwise set $k_{i+1} = k_i$ and find the point $\bar{x}^q$ along the ray λx^q, $\lambda \geq 1$, by solving

LP1 with $\hat{x} = x^q$ and $k = k_{i+1}$. Set $\bar{x}^q = \bar{\lambda}x^q$, where $\bar{\lambda}$ is the solution to LP1. Go to (7).

(6) Let $Y^0(k_{i+1}) = \{x|\phi(x,y)\geq k_{i+1}$ for each $y\epsilon Y_0\}$. For each extreme point $\bar{x}^j$ of P_i, find the intersection of the ray $\lambda\bar{x}^j$, $\lambda \geq 1$, with $Y^0(k_{i+1})$ by solving LP1 with $\hat{x} = \bar{x}^j$ and $k = k_{i+1}$. Redefine $\bar{x}^j$ corresponding to the optimal value $\bar{\lambda}$ of LP1: $\bar{x}^j = \bar{\lambda}\bar{x}^j$. S_i and P_i are defined with respect to these new points $\bar{x}^j$. Go to (7).

(7) For each $\theta_j \in S_i$ (including the one just removed) express $\bar{x}^q$ as a linear combination of the elements of θ_j : $\bar{x}^q = B_j\lambda^j$ or $\lambda^j = B_j^{-1}\bar{x}^q$, with $B_j = [\bar{x}^{j_1},\bar{x}^{j_2},\ldots,\bar{x}^{j_m}]$. Find $e\lambda^j$, where $e = [1,1,\ldots,1]$.

(8) Classify each $\theta_j \in S_i$ (including the one just removed) and $\hat{H}_1,\hat{H}_2,\ldots,\hat{H}_m$ into three mutually exclusive subsets:

a. $L_1 = \{\theta_j \epsilon S_i | e\lambda^j<1\} \cup \{\hat{H}_1,\ldots,\hat{H}_m\}$.

b. $L_2 = \{\theta_j \epsilon S_i | e\lambda^j>1\}$.

c. $L_3 = \{\theta_j \epsilon S_i | e\lambda^j=1\}$.

(9) Find all pairs of elements θ_j and θ_k with $\theta_j \in L_1$ and $\theta_k \in L_2$ which have $(m-1)$ elements in common:
$\bar{x}^{j_i} = \bar{x}^{k_i}$, $i = 1,\ldots,(m-1)$. Define:

$$L_4 = \{(\theta_j,\theta_k)|\theta_j \epsilon L_1,\theta_k \epsilon L_2,\theta_j \text{ and } \theta_k \text{ have } (m-1) \text{ common elements}\}.$$

$$L_5 = \{\theta_e|\theta_e = \text{aff}\left[\text{conv}[\theta_j \cap \theta_k,\bar{x}^q]\right],(\theta_j,\theta_k)\epsilon L_4\}.$$

$$L_6 = \{\theta_m|\theta_m = \text{aff}\left[\text{conv}[\theta_n,\bar{x}^q]\right]\theta_n \epsilon L_3\}.$$

Then

$$S_{i+1} = L_1 \cup L_5 \cup L_6. \quad P_{i+1} = \text{conv}[P_i \cup \{\bar{x}^q\}].$$

Go to (3).

Finiteness of the procedure is easily established. Each time steps (3) through (9) are traversed, one of two things happens: (i) an extreme point of X_0 not contained in P_i (and hence not contained in any of the polytopes generated earlier) is located, (ii) an element from the finite set S_i is removed. Since both can happen only a finite number of times, the algorithm is finite.

5.6 Comparison With Other Algorithms

Several different versions of this algorithm have been proposed in the literature. One of the earliest was by Tui [46]. The problem considered in [46] is of the form:

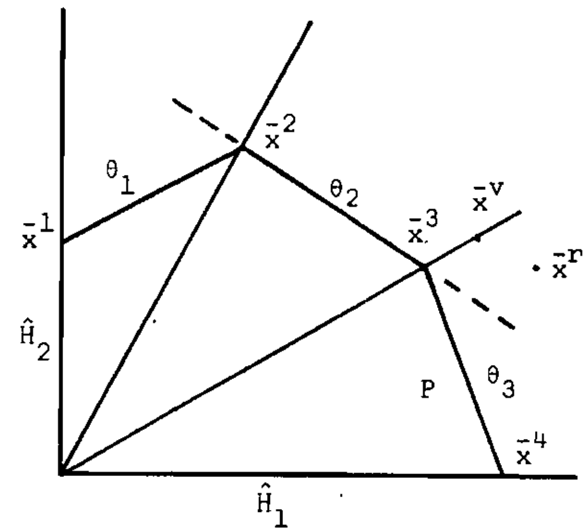
$$\begin{aligned} &\text{Minimize} \quad f(x) \\ &\text{Subject to} \quad x \in D \subset R^n, \end{aligned}$$

where D is a polyhedron and f is concave on D. Concavity of f ensures that a finite minimum, if it exists, will be obtained at an extreme point of D. But there exist local star minima different from the global minimum. In Tui's procedure, the extreme point $\bar{x}^v$ found at any stage is expressed as a linear combination of the elements of θ_j: $\bar{x}^v = \alpha_1 \bar{x}^{j_1} + \ldots + \alpha_m \bar{x}^{j_m}$. For each $\alpha_i \neq 0$ a new set of hyperplanes is defined as follows:

$$H_i = \mathrm{aff}[\bar{x}^{j_1}, \ldots, \bar{x}^{j_{i-1}}, \bar{x}^v, \bar{x}^{j_{i+1}}, \ldots, \bar{x}^{j_m}]$$

Each such hyperplane is added on to the list S_i. Tui conjectured that this procedure will end in a finite number of steps. An example demonstrating that this does not, indeed, happen is given in [50]. The essential problem is that the extreme point $\bar{x}^v$ that is located may be a point of the current polytope P_i. Thus a number of hyperplanes are generated which had already been examined, and this leads to cycling, the same extreme points and hyperplanes being generated again and again.

Tui does not explain why he selects nonzero α_i to generate new hyperplanes. It seems to be from the following consideration. Suppose the polytope P is as shown below.



Let $\bar{x}^v$ be the extreme point found at the *ith* stage. Now $\bar{x}^v = \alpha_3\bar{x}^3 + 0\bar{x}^2$. At this stage, Tui would have generated only one hyperplane $\text{aff}[\bar{x}^2,\bar{x}^v]$. Clearly $\text{aff}[\bar{x}^3,\bar{x}^v]$ is not a desirable hyperplane to generate, since a point $\bar{x}^2$ already contained in P can be the next point located, which can lead to cycling. Unfortunately, simply selecting nonzero α_i is not sufficient to prevent cycling.

It will be instructive to view Tui's algorithm as being one involving generation of a sequence of cones. If $\bar{x}^1$ and $\bar{x}^4$ are the points found along the rays of X_0' incident on 0, then the initial cone is generated by $\{\lambda\bar{x}^1, \lambda\bar{x}^4, \lambda\geq 0\}$. The hyperplane corresponding to this cone is aff$[\bar{x}^1, \bar{x}^4]$. Suppose the point found beyond this hyperplane is $\bar{x}^3$. There are now two cones whose generators are $\{\lambda\bar{x}^1, \lambda\bar{x}^3\}$ and $\{\lambda\bar{x}^3, \lambda\bar{x}^4\}$ such that the union of these two cones is equal to the original cone. The corresponding hyperplanes are aff$[\bar{x}^1, \bar{x}^3]$ and aff$[\bar{x}^3, \bar{x}^4]$. For convergence, it is necessary that each point found must be in the cone corresponding to the hyperplane, that is each α_i must be nonnegative. This need not happen in Tui's algorithm. In the figure above, if $\bar{x}^r$ is found beyond aff$[\bar{x}^2, \bar{x}^3]$, we can see that $\bar{x}^r$ is not an element of the cone $\{\lambda\bar{x}^2, \lambda\bar{x}^3, \lambda\geq 0\}$. The hyperplane aff$[\bar{x}^3, \bar{x}^r]$ could conceivably generate the point $\bar{x}^2$ which is already in P.

One modification suggested is to define new hyperplanes for each $\alpha_i > 0$ instead of each $\alpha_i \neq 0$. Suppose we express $\bar{x}^r = \alpha_2\bar{x}^2 + \alpha_3\bar{x}^3$. Then $\alpha_3 > 0$ but $\alpha_2 < 0$. According to this rule, the only hyperplane generated at this stage is aff$[\bar{x}^r, \bar{x}^2]$ and not aff$[\bar{x}^r, \bar{x}^3]$, which will clearly lead to cycling, $\bar{x}^2$ being beyond it. However, even this modification is not successful because, with reference to the above example, while we know that $\bar{x}^3$ will not reoccur at this stage, we cannot assert that any other extreme point of the current polytope will also not occur. In other words, our knowledge of which points can and cannot reoccur is restricted to the points we are currently working with; we cannot make any statements about extreme points two stages or more away. We will show this by an example later.

The reason why cycling occurs is that the new extreme point $\bar{x}^v$ found at some stage may not be an element of the cone under consideration, i.e. some α is negative. In trying to find the point furthest from $\text{aff}\left[\text{conv}[\bar{x}^2, \bar{x}^3]\right]$, we located the point $\bar{x}^r$ but $\bar{x}^r$ is not in the cone formed by the rays $\{\lambda\bar{x}^2, \lambda\bar{x}^3\}$. Unfortunately, imposing the condition that we find a point x^v as an element of the cone under consideration makes the problem a lot more difficult to solve.

Zwart [49] adds additional constraints to ensure that the new point found is in the desired cone. The effect of adding new constraints is that the new point found need not be an extreme point of X_0. One can then prove convergence to an ε-optimal solution only. An alternative approach may be to require that the new point be both an extreme point of X_0 and in the desired cone. This is an "extreme point mathematical programming problem," for which solution procedures are available, see for example, [25]. While this guarantees finite convergence, it is not clear whether this will be an effective solution procedure computationally. Let us now look at two alternatives that have been proposed to Tui's procedure.

The essential ideas of Tui's procedure were used by Shachtman [39] in the context of a problem in decision theory. In the proof of convergence, it is asserted that a "new" extreme point not in the current polytope is located at each step. If that is so, since the number of extreme points is finite, the algorithm will end in a finite number of steps. However, it appears that new extreme points are not always obtained [40]. In writing a computer program for the algorithm, sufficient bookkeeping was done to ensure that "old" hyperplanes were not added on to the set S_k, that is, a list was maintained of all hyperplanes generated at all stages, and if a hyperplane is generated for the second time, problem LP_2 is not to be solved corresponding to this hyperplane [40].

A third procedure for inductively constructing a polytope is given in [18]. We introduced this method in Chapter I. The optimization problem to be solved is:

$$\text{Maximize} \quad \{c^t x + f^t u \mid (x,u) \epsilon R\}$$

where

$$R = \{(x,u) \mid (x,u) \epsilon P,\ f^t u \le f^t u' \text{ for all } (x,u') \epsilon P\}, \text{ and}$$

$$P = \{(x,u) \mid Ex \le e,\ F^t u \ge d + C^t x, (x,u) \ge 0\}.$$

Though the objective function is linear, the set R may be a nonconvex set. To get around the difficulty, Gallo and Ülkücü work only implicitly with the set R, the real set of interest being X_0. The polytope is constructed using the extreme points of X_0. To start off, a special extreme point x^0 of X_0 is found and the corresponding point u^0, with $f^t u^0 \le f^t u$ for all u such that $F^t u \ge d + c^t x^0$, $u \ge 0$. The point (x^0, u^0) satisfies the condition that $c^t x^j + b^t u^j > c^t x^0 + b^t u^0$ for all $x^j \epsilon N(x^0)$ and $(x^j, u^j) \epsilon R$. Then over the polytope defined by conv$[x^0, \{x^j \epsilon N(x^0)\}]$, the maximum of the objective function is known. Each extreme point x^j is projected along the ray $x^0 + \lambda(x^j - x^0)$, $\lambda \ge 0$, to the point $\bar{x}^j$ such that $c^t \bar{x}^j + f^t \bar{u}^j$ is not greater than the current global maximum. This is obtained by solving a parametric problem in u.

These projections cause the polytope to be enlarged, exactly as in Tui's method. If X_0 is a subset of this polytope, then the global maximum has been found. This question is answered by finding an extreme point of X_0 furthest from $\text{aff}\left[\text{conv}[\{\bar{x}^j\}]\right]$. If x^v is the new point found, then x^v is expressed as a linear combination of the $\bar{x}^j$, and additional hyperplanes are generated by replacing $\bar{x}^j$ with x^v if its coefficient in the linear expression is *positive*. When no more hyperplanes remain, the algorithm terminates. Computationally, the method is no different from directly applying Tui's method to the original problem without taking the dual of the problem in y.

Gallo and Ülkücü [18] "prove" convergence by asserting that no hyperplanes can be repeated if new hyperplanes are generated corresponding to the rule positive coefficients only. We have applied their rule to the following example from Zwart [50], and the conclusion is that hyperplanes can be repeated, so that their algorithm can cycle.

$$\begin{array}{llrcrcrcl}
\text{Minimize} & f(x) = -x_1^2 - x_2^2 - (x_3-1)^2. & & & & & & & \\
\text{Subject to} & & & & -x_2 & & & \le & 0 \\
& & x_1 & + & x_2 & - & x_3 & \le & 0 \\
& & -x_1 & + & x_2 & - & x_3 & \le & 0 \\
& & 12x_1 & + & 5x_2 & + & 12x_3 & \le & 22.8 \\
& & 12x_1 & + & 12x_2 & + & 7x_3 & \le & 17.1 \\
& & -6x_1 & + & x_2 & + & x_3 & \le & 1.9
\end{array} \qquad (1)$$

Let $X_0 = \{(x_1,x_2,x_3) | (x_1,x_2,x_3) \text{ satisfies } (1)\}$.

A local optimum occurs at the vertex $x^0 = (0,0,0)$ with $f(x^0) = -1$. Adjacent vertices are $x^1 = (0.9,0,0.9)$, $x^2 = (0,0.9,0.9)$ and $x^3 = (-0.27,0,0.27)$. A search along each ray yields the projected points $\bar{x}^1 = (1,0,1)$, $\bar{x}^2 = (0,1,1)$ and $\bar{x}^3 = (-1,0,1)$. The first problem to be solved is

$$\text{Maximize} \quad [1,1,1]\begin{bmatrix} 1 & 0 & -1 \\ 0 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix}^{-1}\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

$$\text{Subject to } (x_1,x_2,x_3) \in X_0.$$

The solution is $x^4 = (0,0,1.9)$ and the projected point $\bar{x}^4 = (0,0,2)$. Expressing $\bar{x}^4$ as a linear combination of $\bar{x}^1,\bar{x}^2,\bar{x}^3$ we get

$$\begin{bmatrix} 0 \\ 0 \\ 2 \end{bmatrix} = {}^{1}\begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} + {}^{0}\begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix} + {}^{1}\begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix}$$

Since α_1 and α_3 are positive the set of hyperplanes that exist at this stage are the following:

$$\{\text{aff}\left[\text{conv}[\bar{x}^4,\bar{x}^2,\bar{x}^3]\right]; \quad \text{aff}\left[\text{conv}[\bar{x}^1,\bar{x}^2,\bar{x}^4]\right]\}$$

The calculations are summarized in the following table.

Table 1. Counter-Example to Algorithm of Gallo and Ülkücü

Stage	Hyperplane Defined Through			New Point $\bar{x}^e$	$\bar{x}^e=\alpha_i\bar{x}^i+\alpha_j\bar{x}^j+\alpha_k\bar{x}^k$		
	$\bar{x}^i$	$\bar{x}^j$	$\bar{x}^k$		α_i	α_j	α_k
1	(1,0,1)	(0,1,1)	(-1,0,1)	(0,0,2)	1	0	1
2	(0,0,2)	(0,1,1)	(-1,0,1)	(.055,.66,1.75)	.572	.66	-.055
2	(1,0,1)	(0,1,1)	(0,0,2)	(.055,.66,1.75)	.055	.66	-.518
3	(.055,.66,1.75)	(0,1,1)	(-1,0,1)	None		-	
3	(0,0,2)	(.055,.66,1.75)	(-1,0,1)	None		-	
3	(.055,.66,1.75)	(0,1,1)	(0,0,2)	(-1,0,1)	-18.182	12	10.409
3	(1,0,1)	(.055,.66,1.75)	(0,0,2)	None		-	
3	(1,0,1)	(0,1,1)	(.055,.66,1.75)	None		-	
4	(.055,.66,1.75)	(-1,0,1)	(0,0,2)				
4	(.055,.66,1.75)	(0,1,1)	(-1,0,1)				

We can see that the first hyperplane generated at the fourth stage is the same as the second hyperplane generated at the third stage, and the second hyperplane generated at the fourth stage is the same as the first hyperplane of the third stage. Hence we conclude that hyperplanes can be repeated. It is true that the example considered does not have a bilinear objective function, but we observe that the nature of the objective function plays no role in the inductive construction of the polytopes. All that we need in order to apply the method is the property that the optimal solution will be at an extreme point. We have pointed this out to the authors, but have received no response.

As noted above, the basic difficulty in the algorithms discussed above is cycling. This is essentially because new extreme points (or equivalently new hyperplanes) are not always obtained. This basic difficulty can be overcome by having all the facets of the polytope under consideration, as is done in our algorithm. The points added are always new (i.e. not in the polytope under consideration) and hence we expand the polytope as the method proceeds. We have given the method of finding all the facets of the new expanded polytope.

5.7 Computational Considerations

It is recognized that the algorithm developed in this chapter will require a considerable amount of computer memory and the ability to process lists for realistic problems. A list corresponding to the set S_i will have to be stored defining the current set of facets θ_j and which points are included in each facet. The facets $\hat{H}_j$ corresponding to the coordinate hyperplanes have also to be stored. After the facets have been classified into the sets L_1, L_2, and L_3, the sets L_4, L_5 and L_6 have to be determined by list processing to find out appropriate pairs of facets with (m-1) elements in common. As far as computational burden goes, problem LP 1 is a parametric linear programming problem, a solution procedure for which has already been discussed in Section 4.4. Problem LP3 is a straightforward linear programming problem. It is felt that a good starting solution will be helpful in reducing the amount of computations that will have to be done.

The main difficulty would come in problem LP2, where we need the inverse of a matrix before the problem can be solved. The number of inverses that will need to be evaluated before the procedure terminates depends on when the set S_i becomes empty. One way of reducing the number of inverses to be calculated is to store an appropriate set of inverses, which can be used to generate a much larger set of inverses. We can see that a number of inverses correspond to matrices which differ from a given matrix in precisely one column. This happens when an extreme point $\bar{x}^v$ replaces a number of extreme points of some θ_j to generate a number of facets. Knowing the inverse corresponding to θ_j, one can readily generate the inverses corresponding to the new facets generated. Obviously, there is a trade-off between memory requirements and the computational burden.

Apart from the cutting plane and extreme point ranking algorithms, the performance of this algorithm can be compared with another finite but brute force approach to solving the bilinear problem: that of generating all the vertices of one of the polytopes, say X_0, and solving a linear program over Y_0 for each vertex of X_0. There exist algorithms for finding all vertices of convex polyhedral sets, see for example [5,9], but it has not been established conclusively how effective they are computationally.

5.8 Application

Apart from the fact that the algorithm that we have proposed above is one finite way of solving the bilinear problem, there is one possible application known to us where the optimization problem is defined over a polytope P, where P is not defined in terms of inequalities but as the convex hull of a finite set of points [39]. In this case, our algorithm can be readily applied.

The specific application that has been described in [39] is as follows. Consider a decision-analysis model described by a decision tree. Each path through the tree describes a possible stragegy S^j, $j = 1,\ldots,n$, and has been assigned a certain utility value given a particular state of nature θ_i, $i = 1,\ldots,m$, i.e. let $v_{ij} = V(S^j|\theta_i)$ denote the utility value for strategy S^j when θ_i is the true state of nature. The vector $V^j = (v_{1j},\ldots,v_{mj})^t$ represents m possible values for selecting strategy S^j, the vector of joint conditional values. If $S = \{V^j, j=1,\ldots,n\}$ represents the set of values for all strategies, then conv[S] is the set of all mixed strategies. The objective is to maximize the expected utility $EU = \sum_{i=1}^{m} p_i u_i$ over the polytope conv[S], where $p = [p_1,\ldots,p_m]^t$ is a prior distribution.

Since S is a finite set, one way of maximizing EU is to evaluate the expected utility for each strategy given the vector p of prior distribution. Often the vector p is not known deterministically, so that it will be helpful to be able to do a sensitivity analysis over a range of values for p. In order to do so, it would be useful to define conv[S] in terms of its facets. The method suggested in [39] for doing this is precisely Tui's procedure. We have seen that Tui's algorithm does not converge. Our method for inductively constructing a polytope can be used.

Let the set of facets of the current polytope P_i be $\theta_1,\ldots,\theta_k$. For each facet θ_j, we will express the points v^j remaining in the list $S_i = \{v^j\}$ at the *ith* stage as a linear combination of the elements of θ_j : $v^j = \sum_{i=1}^{m} \lambda_i x^i$. If $\sum_{i=1}^{m} \lambda_i \le 1$, then v^j is removed from S_i and $P_{i+1} = P_i$. If for some v^r, with $v^r = \sum_{i=1}^{m} \alpha_i x^i \sum \alpha_i > 1$, then v^r is removed from S_i and $P_{i+1} = \mathrm{conv}[P_i \cup \{v^r\}]$ and all the facets of P_{i+1} are constructed as shown before. This will clearly be a finite procedure.

5.9 Summary

In this chapter, we have developed an algorithm for inductively constructing a polytope, and indicated some areas where the method has a direct application, in addition to being a tool for solving BLP 1. We have shown that the algorithms proposed in the literature for doing the same may fail to converge, while ours is a finitely convergent algorithm.

CHAPTER VI

DEVELOPMENT OF A LOWER BOUND

6.1 Introduction

In this chapter we will develop two different methods for computing a lower bound for the minimum value of the objective function of BLP 1. In both cases, the bound can be computed without extensive computations. The objective behind computing a lower bound is to terminate the algorithms in case a feasible solution is found at which the objective function value is as close to the bound as desired, or, in case the cutting plane algorithm is used, the lower bound on the remaining feasible region is not less than the current minimum.

6.2 Motivation for Computation of Bounds

An essential ingredient of the cutting plane algorithm presented in this research was to approximate the feasible polytope by the cone formed by the rays incident on an extreme point. Preliminary computational experience reported in [51] indicates that in general as the dimension of the problem increases, the approximation of the feasible set by the cone becomes poor. A consequence of this is that a large number of iterations may be required for cutting plane algorithms because the cuts generated are not very deep. An interesting conjecture is made in [33] and confirmed in [49] on the basis of some computational results. It is hypothesized that the global minimum to a nonconvex minimization problem is usually obtained in the early stages of the implementation of the algorithm, but since there is no mechanism for recognizing this, most of the computational effort is expended in verifying that there does not exist a better solution. If a relatively accurate lower bound on the objective function value were easily obtainable, one could terminate the algorithms whenever a feasible solution close enough to the bound was obtained. Hopefully, a good solution would be obtained in the early stages of the algorithms, specially if Algorithm B discussed in Chapter III is used to get an initial feasible solution. In such cases, one could terminate rather quickly, depending on the level of accuracy desired. We can thus see that obtaining a good lower bound can make the algorithms practicable even for large problems. The properties that the bound must possess are first, it should be close to the true global minimum, and secondly, it should be easy to compute. It seems logical to expect that there will be a trade-off between these properties in the sense that a bound which is easier to compute will be further away from the true minimum. The big problem is that it is usually very hard to say precisely how far away the bound is from the true minimum. Let us now see how a bound for the bilinear problem can be computed.

6.3 Development of First Lower Bound

The basic ideas of this method are an extension of those presented in [10]. Let us restate the problem

$$\text{BLP 1: Minimize} \quad \phi(x,y) = c^t x + d^t y + x^t Cy$$
$$\text{Subject to} \quad x \in X_0 = \{x \mid Ex \le e,\ x \ge 0\}$$
$$y \in Y_0 = \{y \mid Fy \le f,\ y \ge 0\}.$$

Let c^i be the *ith* row of the matrix C. Let u_i be the optimal value of the objective function of the following linear programming problem:

$$\text{Minimize} \quad (c^i)y$$
$$\text{Subject to} \quad y \in Y_0$$

Thus $u_i \le c^i y$ for all $y \in Y_0$, $i = 1,\ldots,m$. Since $x \ge 0$, we have $x^t u \le x^t Cy$ for all $y \in Y_0$, where $u^t = [u_1,\ldots,u_m]$. Thus, $c^t x + d^t y + x^t u \le c^t x + d^t y + x^t Cy$ for all $y \in Y_0$. We now see that the variables x and y have been separated. We now define the following problems:

$$L_1^j : \text{Minimize } (c^t + u^t)x,\ x \in X_0^j$$

$$L_2 : \text{Minimize } d^t y,\ y \in Y_0$$

where $X_0^j \subset X_0$. We will discuss below how X_0^j can be specified. Let z_1^j and z_2 be, respectively, the minimum value of the objective function of L_1^j and L_2. Now $z_1^j + z_2 \le c^t x + u^t x + d^t y \le c^t x + x^t Cy + d^t y$ for all $x \in X_0^j$ and $y \in Y_0$.

One way of implementing this bound is to set $X_0^j = X_0$, that is, to evaluate $z_1 + z_2$ before any cutting planes have been added, and use this as a fixed lower bound. Alternatively, the set X_0^j is obtained from X_0^{j-1}

by adding a cutting plane at the j*th* stage. Since $X_0^j \subset X_0^{j-1}$, $z_1^{j-1} \le z_1^j$. Hence $z_1^{j-1} + z_2 \le z_1^j + z_2$. Thus at each stage we have an improved lower bound on the remaining feasible region of X_0. If the current global minimum is less than the lower bound on the remaining feasible region, the algorithm terminates.

The proximity of the bound to the true global minimum depends on the relative magnitudes of the linear and the bilinear terms. If the elements of C are much smaller than the elements of c and d it would be reasonable to expect the bound to be a fairly good approximation. In other cases, it would probably depend on the form of the matrix C. For the numerical example presented in Chapter IV, the lower bound, computed as indicated above, turns out to be 10, which is precisely equal to the true global minimum. The pseudo-global minimum is 12.

6.4 Development of Second Lower Bound

Let $\bar{x} \in X_0$ be a vertex such that $(\bar{x},\bar{y})$ is a pseudo-global minimum obtained according to Algorithm A of Chapter III. Let J be the set of nonbasic variables at $\bar{x}$. Let $P = \{(x^t u^t)^t | Ex+u=e,\ x\ge 0,\ u\ge 0\}$. Suppose the cutting plane introduced according to the cutting plane algorithm of Chapter IV is $\sum_{j\in J} p_j/\bar{\lambda}_j \ge 1$, where $p \in P$. We will first show how to define the hyperplane $H_b = \{x | x = \bar{x} - \sum_{j\in J} \bar{e}^j p_j,\ \sum_{j\in J} p_j/\bar{\lambda}_j = b\}$ such that $X_0 \subset H_b^-$.

The hyperplane H_b is the translate of the cutting plane $H = \{x | x = \bar{x} - \sum_{j\in J} \bar{e}^j p_j,\ \sum_{j\in J} p_j/\bar{\lambda}_j = 1\}$. Let p be defined such that $\bar{x} \in H^-$.

Let $S_1 = \{x | x = \bar{x} - \sum_{j\in J} \bar{e}^j p_j,\ p_j \ge 0,\ j\in J\}$, which by reference to Section

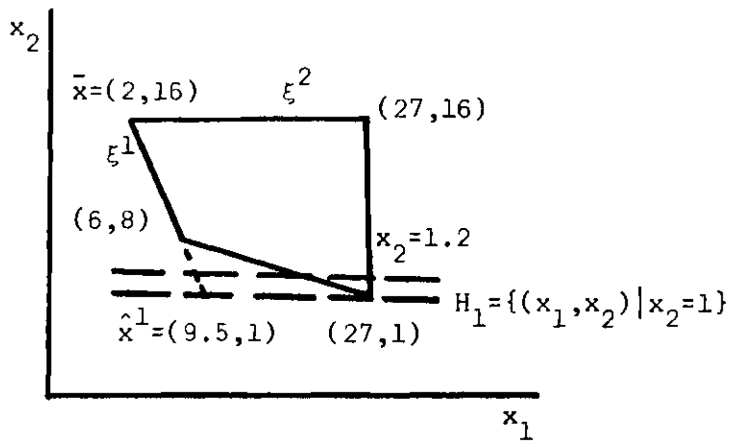
3.3, is the set obtained from X_0 by deleting the nonnegativity restrictions on the basic variables at the vertex $\bar{x}$ and hence $X_0 \subset S_1$. Now $H^- \cap S_1$ is a polyhedron with vertices $\bar{x}$ and $\bar{x} - \bar{e}^j\bar{\lambda}_j$, $j \in J$, whenever $\bar{\lambda}_j < \infty$, and whose extreme directions are $-\bar{e}^j$, $j \in J$, whenever $\bar{\lambda}_j = \infty$ (see proof of Theorem 4.6). Also, along any extreme direction of $H^- \cap S_1$, Min $\phi(\bar{x}-\bar{e}^j\lambda,y) \geq \phi(\bar{x},\bar{y})$ for all $\lambda \geq 0$ and $y \in Y_0$. $H_b^- \cap S_1$ is also a polyhedron with vertices $\bar{x}$ and $\bar{x} - \bar{e}^j\bar{\lambda}_j b$, $j \in J$ whenever $\bar{\lambda}_j < \infty$ and with extreme directions the same as those of $H^- \cap S_1$. Hence if we enumerate $\underset{y\in Y_0}{\text{Min}}\ \phi(x,y)$ for all extreme points of $H_b^- \cap S_1$, we will get $\underset{y\in Y_0}{\text{Min}}\ \phi(x,y)$ over $x \in H_b^- \cap S_1$. But X_0 is contained in both H_b^- and S_1. Hence we get a lower bound for Min $\phi(x,y)$ for all $x \in X_0$, $y \in Y_0$.

To summarize, let $\hat{x}^j = \bar{x} - \bar{e}^j\bar{\lambda}_j b$ for $j \in J_1 = \{j|j\in J, \bar{\lambda}_j < \infty\}$. Let $\bar{Z}$ and Z^j be the minimal value of the objective function for the following linear program for $x = \bar{x}$ and $x = \hat{x}^j$, $j \in J_1$:

$$\begin{array}{ll} \text{Minimize} & \phi(x,y) \\ \text{Subject to} & y \in Y_0 \end{array}$$

Let $Z = \min\ \{\bar{Z}, Z^j,\ j\in J_1\}$. Then Z is a lower bound for Min $\phi(x,y)$ for all $x \in X_0$, $y \in Y_0$.

To illustrate, consider Example Problem I discussed in Section 3.2. The cut introduced at $\bar{x} = (2,16)$ was $x_2 \leq 1.2$ as shown in the figure below.



Then $H_b = \{(x_1,x_2)|x_2=1\}$. The ray ξ^1 is from (2,16) to (6,8) and ray ξ^2 is from (2,16) to (27,16). ξ^2 does not intersect H_b (that is, ξ^2 is an extreme direction) and ξ^1 intersects H_b at $\hat{x}^1 = (9.5,1)$. The minimum of the linear program

$$\begin{array}{ll} \text{Minimize} & x_1y_1 + x_2y_2 \\ \text{Subject to} & (y_1,y_2) \in Y_0 \end{array}$$

is 10 for $(x_1,x_2) = (9.5,1)$ and 12 for $(x_1,x_2) = (2,16)$. Hence the lower bound for $\phi(x,y)$ is $\min\{12,10\} = 10$.

CHAPTER VII

CONCLUSIONS, COMPUTATIONAL RESULTS AND RECOMMENDATIONS

7.1 Conclusions

In this research, we have developed two cutting plane algorithms and one polytope generation algorithm for solving BLP 1. The first cutting plane algorithm is infinitely convergent, and the second converges finitely to an ε-optimal solution. If $\varepsilon = 0$ and a pseudo-global minimum instead of a local star minimum is found at each stage, the two cutting plane algorithms become identical. Furthermore, if $\epsilon > 0$ is small enough, the algorithm is likely to yield a global minimum in a finite number of steps and using only a local star minimum. We will now discuss some of the algorithms proposed in the literature and compare with our algorithms.

Konno [26] has developed a cutting plane algorithm for solving BLP 1. We have proved that our cuts dominate those due to Konno. In Chapter II, we have shown that a concave quadratic problem can be transformed into a bilinear problem. Preliminary computational results have shown that our algorithm is two to five times faster than Moreno's [33].

We have proved infinite convergence of our cutting plane algorithm to a global optimum. In order to do so, we used the local minimum property of the pseudo-global minimum to satisfy condition (4) of the Convergence Theorem 4.9, and continuity of the function Δ.

Table 2. Algorithms for Nonconvex Linearly Constrained Problems

Algorithm	Concave		Nonconvex	BLP	Convergence	Computations Involved	Starting Solution Used
	Quadratic	General	Quadratic				
I HEURISTIC SEARCH							
1. Mueller [34]			X		Need not converge.	Calculate gradient. One dimensional search.	Arbitrary Point
II EXTREME POINT RANKING							
2. Cabot & Francis [10]			X		Finitely convergent (see discussion below).	Rank extreme points of L.P.	Optimal Solution to L.P.
3. Taha [45]		X			Finitely convergent (see discussion below).	Rank extreme points of L.P.	Optimal Solution to L.P.
III CUTTING PLANE							
4. Candler & Townsley [11]			X		No proof available.	Search through extreme points.	Local Minimum
5. Tui I [46]		X			No proof available.	Parametric L.P.	Local Star Min*
6. Moreno [33]		X			No proof available (see discussion below).	Generate all adjacent extreme points. Solve LPs.	Local Star Min*
7. Burdet & Balas [4]	X				Infinitely convergent but no proof given (see discussion).	Parametric L.P.	Local Star Min*
8. Ritter [38]			X		Need not converge (see [50]).	Enumerate all solutions of a system of equations with complementary restrictions.	Local Min
9. Konno [26]				X	Finite convergence to ε-optimal solution (see discussion).	L.P. Parametric L.P.	Local Star
10. Balas [3]			X		Infinitely convergent.	Parametric L.P.	A vertex of Kuhn Tucker conditions satisfying complementary slackness conditions.
11. Vaish I				X	Infinite convergence to global minimum; finite convergence to ε-optimal solution.	L.P. & Parametric L.P.; generation of alternative optimal solutions.	Pseudo-Global, Local Star if ε-optimal solution required.
IV POLYTOPE GENERATION							
12. Tui II [46]		X			Need not converge (see [50]).	L.P. and one dimensional search. Matrix inversion.	Local Star*
13. Zwart [49]		X			Finite convergence to ε-optimal solution.	L.P. and one dimensional search. Matrix inversion.	Local Star*
14. Gallo & Ülkücü [18]				X	Need not converge (see discussion below).	L.P. & Parametric L.P. Matrix inversion.	A point having some special property.
15. Shactman [39]†					Need not converge without special programming aids.	Matrix inversion.	
16. Vaish II				X	Finite convergence to global min.	L.P. and Parametric L.P., Matrix inversion.	Pseudo-Global

*In these cases, a local star is also a local minimum.

†Deals only with the problem of generating the facets of the convex hull of a given finite set of points.

We have also proved finite convergence of our algorithm to an ε-optimal solution. For this purpose, we required that the initial solution be a local star minimum, since it is always an interior point of the polar set $Y^0(k-\varepsilon)$. Thus condition (4) of Theorem 4.9 will always be satisfied.

In order to define a cutting plane in R^m, we need m points. If the pseudo-global minimum $\bar{x}$ is non-degenerate, there are exactly m adjacent extreme points to $\bar{x}$. We can identify m rays incident on $\bar{x}$, find m points of intersection of these rays with the polar set and uniquely define a valid cutting plane. If $\bar{x}$ is degenerate, there can be more than m adjacent extreme points. In this case, it is not clear which points are to be used to define the cutting plane. We have resolved this problem by using the set X_0' obtained from X_0. X_0' has precisely m rays incident on the pseudo-global minimum, and we have shown how to uniquely define a valid cutting plane using these rays.

We will now examine the question of convergence and resolution of degeneracy of the algorithms proposed in the literature for BLP 1 and related problems (see Table 2). Extreme point ranking methods are finitely convergent and there is no problem with degeneracy. However, ranking all extreme points is computationally not very efficient for realistic problems.

For cutting plane algorithms, we have seen in Theorems 4.6 and 4.9 that having a local minimum is important since, otherwise, the current point may not be cut off. The algorithms listed in Table 2 dealing with general concave functions start with a local star minimum. In this case, such a point is in fact a local minimum [33].

Among the cutting plane algorithms, there is one due to Tui. He does not prove convergence of his cutting plane algorithm. Moreover, he states that perturbation techniques can take care of degeneracy. But Moreno [33] has shown that this is not true. Hence it seems unlikely that convergence can be proved for Tui's algorithm. Moreno generates all adjacent extreme points of a degenerate vertex, projects them along the corresponding rays joining them to the local star minimum and finds the best hyperplane by linear programming. However, generating all adjacent extreme points may not be very easy to implement for highly degenerate vertices. Moreno does not address himself to the question of convergence at all. His major objective was "the development of an 'efficient' algorithm" since a finite algorithm--searching all local star minima--already exists.

Burdet and Balas assert that proving finite convergence to a global minimum may not be easy. They have resolved degeneracy by identifying m rays incident on a local star minimum by removing constraints. For these reasons, it is likely that their cutting plane algorithm is infinitely convergent.

Ritter's algorithm is not even infinitely convergent as shown in [50]. However, Konno adopts Ritter's algorithm for a Bilinear Problem and proves finite convergence to an ε-optimal solution. Even though we do not agree with his convergence "proof" it would appear that his conclusion is true. His initial point is a local star minimum which need not be a local minimum. Suppose his cut is given by $g^t x \geq \sigma$. He shows that $\sigma \geq \frac{\phi_p - \phi_0}{K_p}$ where ϕ_0 is the current global minimum, ϕ_p is the p*th* local star minimum, and K_p is a constant determined at the p*th* stage. If $\phi_0 = \phi_p$, then σ could be zero, which means that no part of the feasible region is cut off. This happens because of the absence of the local minimum property. Thus Konno always sets $\phi_p - \phi_0 = \varepsilon > 0$ for some pre-determined ε. Moreover, K_p is a function of the elements of the current basis inverse, B^{-1}. If $K_p \to \infty$, $\sigma \to 0$. Konno asserts that $K_p \to \infty$ if some elements of B^{-1} tend to infinity. This can only happen if a series of parallel cuts are generated. To get around this, his algorithm will generate a cut passing through the adjacent extreme points to the current local star minimum. However, if the local star minimum is an extreme point which was created by adding cutting planes, one cannot assert that its adjacent extreme points can be no closer than a pre-specified distance. Thus it is not clear how K_p is bounded.

The second algorithm that we have developed is a polytope generation procedure. This algorithm converges to a global minimum in a finite number of steps. This is because we generate all the facets of the current polytope. Thus no extreme point is ever repeated. Table 2 lists this type of algorithms proposed in the literature. Tui's polytope generation procedure need not converge as shown by Zwart [50]. The reason is that points already contained in the current polytope can be found once again, which leads to cycling. Zwart [49] showed that a sufficient condition to prevent cycling is to find points in a specified cone. He added extra restrictions which enabled him to do so. However, the algorithm converges to an ε-optimal solution. Gallo and Ülkücü's algorithm need not converge, as we have shown by an example in Chapter V.

Once again, the difficulty is that extreme points already contained in the current polytope are located again. The modification suggested by them is not sufficient to prevent this. Shachtman [39] directly adopted Tui's polytope generation algorithm, which, of course, need not converge. While programming, sufficient record keeping was done to prevent cycling. Thus, insofar as polytope generation procedures are concerned, ours is the only algorithm which converges finitely to a global optimum. From the statement of the algorithm we see that the computations involved in our procedure compare favorably with what is required in similar algorithms.

7.2 Computational Results

The Appendix lists the computer program for the ε-optimal algorithm of Section 4.8. The program can handle problems whose X_0 (or Y_0) set is defined by a matrix of size 50×75 including all cuts added. The code was tested on randomly generated data. Table 4 lists the computational results. Table 3 lists the computational results, again from randomly generated data, obtained by setting ε equal to zero and gives the global optimum. There are no published results to compare with for a bilinear problem.

We have also tested our code on three problems provided by Moreno. The results are shown in Table 5. In one case our computation time is 1/5 that of Moreno, assuming that the Univac 1108 is twice as fast as the CDC 6400. In the other two cases the solution provided by Moreno is not feasible to the problems stated. Our program yielded an optimal feasible solution.

Table 3. Global Optimal Results

Problem Size		Computational Time* (Seconds)
X_0-Matrix	Y_0-Matrix	
4×2	4×2	.36
7×4	5×8	.92
6×9	4×8	1.1
6×8	8×5	1.2
4×7	7×9	1.2
4×7	6×9	1.3
3×5	8×15	1.6
6×9	6×9	1.8
10×12	5×8	2.0
10×15	10×15	3.3

*All times in seconds on Univac 1108 exclusive of input/output time.

Table 4. ε-Optimal Computational Results

Problem Size		Computational Time* (Seconds)
X_0-Matrix	Y_0-Matrix	
4×2	4×2	.36
7×4	5×8	.92
6×9	4×8	1.1
6×8	8×5	1.2
4×7	7×9	1.2
4×7	6×9	1.3
6×8	6×8	1.4
3×5	8×15	1.6
10×12	5×8	2.0
10×15	10×15	3.3

*All times in seconds on Univac 1108 exclusive of input/output time.

Table 5. Times for Two Algorithms

Problem Size*	Computational Time	
	Moreno (on CDC 6400)# [33]	Vaish (on Univac 1108)
5×5†	3.2 Seconds	.249 Seconds
10×10†	4.8 Seconds	2.859 Seconds
10×10	5.0 Seconds	.417 Seconds

*For a concave problem defined over a polyhedral set, the equivalent bilinear problem has two matrices defining the constraint set, each of the same size as the single matrix of the concave problem.

†Moreno's solution [33] to the problem does not satisfy the constraints.

#According to Auerbach Handbook, Univac 1108 is about twice as fast as the CDC 6400.

7.3 Recommendations

The Bilinear Problem has the special property that it reduces to a linear programming problem for a fixed y (or x). For a linear programming problem defined over a polytope an optimum solution is attained at a vertex and a local minimum is a global minimum. We have seen in Chapter III that these two properties hold for a more general class of functions. Quasi-concavity ensures extreme point optimum, and strict quasi-convexity ensures that a local minimum is a global minimum. Thus the cutting plane algorithm can be extended to solve problems whose objective function has these properties for a fixed y (or x). One such example is the *Fractional Bilinear Problem*

$$\text{Minimize} \quad \frac{\phi_1(x,y)}{\phi_2(x,y)}$$

$$\text{Subject to} \quad x \in X_0$$

$$y \in Y_0$$

For a fixed x (or y) this is a fractional programming problem. There exist efficient computational procedures for solving a fractional programming problem [28]. Hence the power of linear programming methods can be extended to this class of problems.

It is well known that an absolute value problem of the form $\text{Min} \sum_{j=1}^{n} c_j |x_j|$, subject to $\sum_{j=1}^{n} a_{ij} x_j \le b_i$, $i = 1,\ldots,m$ can be transformed into a linear programming problem if $c_j \ge 0$. Let us now consider the *Bilinear Absolute Value Problem*

$$\text{Minimize} \quad \sum_{j=1}^{n} c_j |x_j y_j| \qquad (c_j \geq 0,\ j=1,\ldots,n)$$

$$\text{Subject to} \quad \sum_{j=1}^{n} a_{ij} x_j \leq b_i, \quad i = 1,\ldots,m$$

$$\sum_{j=1}^{n} d_{ij} y_j \leq f_i, \quad i = 1,\ldots,m.$$

We observe that for a fixed x_j, $j = 1,\ldots,n$ (or y_j, $j=1,\ldots,n$), this problem reduces to an absolute value problem, which can be solved by linear programming. One can investigate the nature of the initial point with respect to which a cutting plane is to be defined and construct a cutting plane algorithm for finding a global minimum to the problem.

APPENDIX

This appendix lists the computer code for the ε-optimal solution (see Section 4.8). The comment "Test Alternative Optimal Solutions" of the code is not part of the algorithm developed in Section 4.8. It is incorporated since it may prove to be an easy method of getting a better local star optimum. This part of the code finds only the alternative optimum solutions adjacent to $\bar{y}$ (for a fixed $\bar{x}$), where $(\bar{x},\bar{y})$ is a local star optimum.

```
C       THIS PROGRAM SOLVES A PROBLEM OF THE FORM
C       MINIMIZE  CX + DY +XQY
C       SUBJECT TO  EX .LE.U,X .GE. 0, FY .LE.V, Y .GE. 0
C       BY THE CUTTING PLANE ALGORITHM
C       INPUT DATA IS AS FOLLOWS
C       THE FIRST CARD CONTAINS THE NUMBER OF PROBLEMS TO BE
C       SOLVED (I5), THE TOLERANCE LEVEL EPSILON (F10.0),
C       AND THE LARGE NUMBER FOR SOLVING THE PARAMETRIC
C       PROBLEM (F10.0)
C       THE FIRST 12 COLUMNS OF THE NEXT  CARD CONTAIN THE NUMBER
C       OF ROWS AND COLUMNS IN E, AND THE NUMBER OF ROWS AND
C       COLUMNS IN F  (4I3)
C       THE NEXT SET OF CARDS CONTAIN THE ELEMENTS OF THE
C       MATRICES E, F AND Q ARRANGED ACCORDING TO ROWS
C       AND IN THE FORMAT 5F10.0
C       THE DATA FOR EACH PROBLEM TO BE SOLVED IS ENTERED
C       AS ABOVE
      DIMENSION EX(50,75),EY(50,75),ERX(50),ERY(50),COX(75),COY(75)
     2 COXY(75,75),X(75), B(50,50),SU(50),TEM(75),YY(75)
     3EX1(50,75),EY1(50,75),CURMIN(75),OBAR(75),ABAR(60),TEM2(60)
     4INDIC(190),IB(50),TEM3(110),BX(50,50),INDI(180),IB3(50)
     5AT(50),CT(75),EY2(75,75)
       COMMON B,IB,INDIC,CT,AT
       INST=0
      READ (5,9641) NUM,EPS,SL
 9923 CONTINUE
       ZMIN = 99999.0
      READ (5,621) MX,NX,MY,NY
 9641 FORMAT (I5,2F10.0)
      DO 371 I=1,MX
 371  READ (5,622) (EX(I,J), J=1,NX)
       DO 372 I=1,MY
 372   READ (5,622) (EY(I,J), J=1,NY)
       DO 373 I=1,NX
 373   READ (5,622) (COXY(I,J),J=1,NY)
      READ (5,622) (ERX(J),J=1,MX)
      READ (5,622) (ERY(J),J=1,MY)
      READ (5,622) (COX(J),J=1,NX)
      READ(5,622) (COY(J),J=1,NY)
 622  FORMAT (5F10.0)
 621  FORMAT (4I3)
      MSTAR = MX
      MSTAR1 = MSTAR + 1
      KKK = 0
      MY1 = MY + 1
      LY = 0
      DO 920 I = 1,MY
      IF(ERY(I).GE.0.0) GO TO 920
      LY = LY + 1
 920  CONTINUE
      LZ = LY + NY
      LK = LY + NY + MY
      LZ1 = LZ +1
      WRITE (6,891)
 891  FORMAT (1H1,5X,'OBJECTIVE FUNCTION VALUE AT LOCAL STAR MIN ',//)
 824  MX1 = MX+1
      K1K = 0
      DO 842 I=1,MX
      IF (ERX(I).GE.0.0) GO TO 842
```

```
      K1K = K1K + 1
  842 CONTINUE
      K4K = K1K + NX
      K2K = K1K + MX + NX
      K4K1 = K4K+1
C      FIND INITIAL SOLUTION TO X-SET
      DO 605 I = 1,NX
  605 EX(MX1,I)=0.
      NX1 = NX
      DO 625 I = 1,MX1
      DO 625 J = 1,K4K1

  625 EX1(I,J) =EX(I,J)
      CALL LINP(EX1,MX,NX1,ERX,X,SU,Z,IS)
C      TERMINATE IF NO FEASIBLE SOLUTION
      IF (IS.EQ.-1.OR.IS.EQ.1) GO TO 691
C      FIX X AND SOLVE AN LP IN Y
  611 DO 603 K = 1,NY
      TEM(K) = 0.
      DO 603 I = 1,NX
  603 TEM(K) = TEM(K) + X(I)*COXY(I,K)
      DO 604 I = 1,NY
  604 EY(MY1)=COY(I)+TEM(I)
      NY1 = NY
      DO 626 I=1,MY1
      DO 626 J=1,LZ1
  626 EY1(I,J) = EY(I,J)
      CALL LINP(EY1,MY,NY1,ERY,YY,SU,ZY,ISY)
      IF(ISY.EQ.-1.OR.ISY.EQ.1) GO TO 691
C      FIX Y AND SOLVE AN LP IN X
      DO 606 K = 1,NX
      TEM(K) = 0.
      DO 606 I = 1,NY
  606 TEM(K) = TEM(K) + YY(I)*COXY(K,I)
      DO 608 I = 1 ,NX
  608 EX(MX1,I) = COX(I) + TEM(I)
      NX1 = NX
      DO 627 I = 1 ,MX1
      DO 627 J = 1 ,K4K1
  627 EX1(I,J) = EX(I,J)
      CALL LINP(EX1,MX,NX1,ERX,TEM,SU,Z,IS)
C      IF NO CHANGE IN X, A LOCAL STAR IS OBTAINED
      DIFF = 0.
      DO 609 I =1,NX
  609 DIFF = DIFF + ABS(X(I)-TEM(I))
      IF (DIFF.LE.0.00005) GO TO 612
       DO 610 I = 1,NX
  610 X(I) = TEM(I)
      GO TO 611
  612  CONTINUE
       DO 531 K= 1,NY
      TEM(K) = 0.
       DO 531 I = 1,NX
  531 TEM(K) = TEM(K) + X(I)*COXY(I,K)
       DO 532 I=1,NY
  532 EY(MY1,I) = COY(I) + TEM(I)
       NY1 = NY
       DO 538 I = 1 , MY1
       DO 538 J = 1 , LZ1
```

```
538   EY1(I,J) = EY(I,J)
       CALL LINP (EY1,MY,NY1,ERY,YY,SU,ZY,ISY)
      IF(ZMIN.GT.ZY)ZMIN=ZY
       DO 534 I = 1,NY
534    CURMIN(I) = YY(I)
       DO 536 I = 1,NY
       DO 536 J =1,NY
536    EY2(I,J) = 0.
       KP=0
C     TEST ALTERNATIVE OPTIMAL SOLUTIONS
       DO 500 KJ = 1,LK
       IF (INDIC(KJ) .NE.0) GO TO 500
       IF (KJ .GT. LZ) GO TO 503
       IF (ABS(CT(KJ)).GE.0.005) GO TO 500
       KP = KP +1
       IX=KJ
       DO 501 I=1,MY
      AT(I)=0.0
       DO 501 J=1,MY
501   AT(I) = AT(I) + B(I,J)*EY1(J,IX)
       AT(MY1)= 0.
       GO TO 502
503   IF(ABS(B(MY1,KJ)).GE.0.005) GO TO 500

        KP = KP+1
       IX = KJ
       DO 504 J = 1,MY1
504    AT(J) = B(J,IX)
502    CONTINUE
       KS =0
       DO 505 I=1,MY
       IF (AT(I).LE.0.005) GO TO 505
       IF(KS.NE.0) GO TO 506
      IR = I
      BM = B(I,MY1)/AT(I)
       KS = 1
       GO TO 505
506   BZ = B(I,MY1)/AT(I)
       IF (BZ-BM) 507,508,505
507    KZZ=0
      BM=BZ
       IR=I
       GO TO 505
508    KZZ =KZZ+1
505    CONTINUE
       IIIX =IB(IR)
       INDIC(IIIX) =0
       INDIC(KJ)=1
       IB(IR) =KJ
       DO 510 J=1,MY
       IF(J.EQ.IR) GO TO 510
      TEM4(J)=B(J,MY1)+(AT(J)*B(IR,MY1))/AT(IR)
510    CONTINUE
      TEM4(IR)=B(IR,MY1)/AT(IR)
      DO 512 J=1,MY
       IF(IB(J).GT.NY) GO TO 512
       JKL = IB(J)
       EY2(KP,JKL)=TEM4(J)
512     CONTINUE
```

```
 500    CONTINUE
        IF ( KP.EQ.0) GO TO 548
        DO 520 KJ = 1,KP
        DO 521 K=1,NX
        TEM(K)=0.
        DO 521 I=1,NY
 521   TEM(K)=TEM(K)+EY2(KJ,I)*COXY(K,I)
        DO 523 I=1,NX
 523   EX(MX1,I)=COX(I)+TEM(I)
        NX1=NX
        DO 524 I=1,MX1
        DO 524 J=1,K4K1
 524    EX1(I,J)=EX(I,J)
        CALL LINP(EX1,MX,NX1,ERX,TEM,SU,Z,IS)
        IF (Z.GT.ZMIN) GO TO 520
         ZMIN=Z
       DO 525 I=1,NX
 525    X(I)=TEM(I)
        GO TO 611
 520    CONTINUE
 548    CONTINUE
        DO 541 K=1,NX
        TEM(K)=0.
        DO 541 I=1,NY
 541   TEM(K) = TEM(K)+ CURMIN(I)*COXY(K,I)
        DO 542 I=1,NX
 542   EX(MX1,I)=COX(I)+TEM(I)
        NX1=NX
        DO 533 I=1,MX1
        DO 533 J=1,K4K1
 533    EX1(I,J)=EX(I,J)
        CALL LINP (EX1,MX,NX1,ERX,TEM,SU,Z,IS)
         WRITE (6,624) Z
 624   FORMAT (15X, F12.6)
C       THE GLOBAL MIN IS IN ZMIN

       IF(Z.LE.ZMIN) ZMIN = Z
       IF(Z.GT.ZMIN) GO TO 827
C       THE CURRENT MIN POINT IS CURMIN
       DO 828 J=1,NX
 828   CURMIN(J) = X(J)
 827   CONS1 = 0.0
       ZMIN = ZMIN-EPS
       DO 808 KL=1,NX
C   FIND C*XBAR
 808   CONS1 = CONS1 + COX(KL)*X(KL)
C  FIND XBAR*COXY AND ADD TO COY
       DO 810 KI=1,NY
       TEM(KI) = 0.0
       DO 811 KII = 1,NX
 811   TEM(KI) = TEM(KI) + X(KII)*COXY(KII,KI)
 810   DBAR(KI) = COY(KI)+TEM(KI)
       DO 834 K3 =1,K2K
 834   TEM3(K3) = 0.0
C  GENERATE ABAR
       DO 871 LL =1,K2K
       INDI(LL) = INDIC(LL)
 871   IB3(LL) = IB(LL)
       DO 850 L1=1,MX1
```

```
      DO 850 L2=1,MX1
850    BX(L1,L2)= B(L1,L2)
      DO 801 IP = 1,K2K
      IF (IP.LE.K4K) GO TO 805
C      TEST IF COL IS OF ARTIFICIAL VARIABLE
       IF (ERX(IP-K4K).LT.0.0) GO TO 801
C      TEST FOR BASIC VARIABLE
805   IF(INDI(IP).EQ.1) GO TO 801
      DO 802 J=1,NX
802   ABAR(J) = 0.0
      IF (IP.LE.NX) ABAR(IP)= 1.0
C      FIND UPDATED COLUMN A(IP)
      IF(IP.GT.K4K) GO TO 851
      DO 803 J = 1,MX
      TEM(J) = 0.0
      DO 803 K= 1,MX
803   TEM(J) = TEM(J)+ BX(J,K)*EX1(K,IP)
      GO TO 852
851   CONTINUE
C      UPDATED COL OF SLACK VARIABLE IS IN B INVERSE
      DO 853 K = 1,MX
853   TEM(K) = BX(K,IP-K4K)
852   CONTINUE
       DO 804 JI = 1,MX
      JKL = IOB(JI)
      IF(JKL.GT.NX) GO TO 804
      ABAR(JKL) = -TEM(JI)
804   CONTINUE
C      FIND -C*-ABAR
      CONS2 = 0.0
       DO 809 J = 1,NX
809   CONS2 = CONS2 + COX(J)*ABAR(J)
C      FIND -ABAR*-COXY
      DO 812 I=1,NY
      TEM(I)=0.0
      DO 812 J=1,NX
812   TEM(I) = TEM(I)+ ABAR(J)*COXY(J,I)
C     SOLVE PARAMETRIC  PROBLEM
      HIL =SL
      SMAL = 0.0
      RLAM = HIL
      K2 = 0
817   CONTINUE
      DO 814 J=1,NY
814   EY(MY1,J) = CBAR(J) + RLAM*TEM(J)
      DO 815 KI = 1,MY1
      DO 815 J = 1,LZ1

815   EY1(KI,J) = EY(KI,J)
      NY1 = NY
      CALL LINP(EY1,MY,NY1,ERY,YY,TEM2,ZY,ISY)
      ZIN = ZY-CONS1-RLAM*CONS2
      IF(K2.EQ.1) GO TO 818
C    IS PARAMETRIC MIN BEYOND LAMDA = SL
      IF (ZIN.GE.ZMIN) GO TO 821
      RLAM = 0.5*(HIL+SMAL)
      K2 = 1
      GO TO 817
818   DIFF =ZIN -ZMIN
```

```
C       IS PARAMETRIC MIN WITHIN EPS OF CURRENT GLOBAL MIN
        IF(DIFF.GE.C.0.AND.DIFF.LE.EPS) GO TO 821
C       REDUCE INTERVAL OF UNCERTAINTY BY 1/2
       IF(DIFF) 816,819,819
 816   HIL = RLAM
       GO TO 820
 819   SMAL = RLAM
 820   RLAM = 0.5*(SMAL+HIL)
       GO TO 817
 821   CONTINUE
C       COEFF OF COL IP IN CUT
       TEM3(IP) = 1.0/RLAM
 801   CONTINUE
C        STORE COEFF OF REAL VARIABLES IN LAST ROW
       DO 876 J=1,NX
 876   EX(MX1,J) = -TEM3(J)
       ERX(MX1) = -1.0
C       CONVERT COEFFS OF SLACKS TO REAL VARIABLES
       DO 872 I=1,MX
       IF (ERX(I)) 873,874,874
 873   DO 875 J =1,NX
       EX(MX1,J) = EX(MX1,J)-TEM3(NX+1)*EX1(I,J)
 875   CONTINUE
       ERX(MX1) = ERX(MX1)-TEM3(NX+1)*ABS(ERX(I))
       GO TO 872
 874   DO 877 J=1,NX
       EX(MX1,J) = EX(MX1,J)+TEM3(K4K+1)*EX1(I,J)
 877   CONTINUE
       ERX(MX1) = ERX(MX1)+ TEM3(K4K+1)*ABS(ERX(I))
 872   CONTINUE
       DO 878  I=1,K2K
 878   TEM3(I) =0.0
       MX = MX + 1
       KKK = KKK + 1
       GO TO 824
C       FIND EXTREME POINTS OF ORIGINAL SET
 691   DO 931 I=1,NY
       TEM(I) = 0.0
       DO 931 J=1,NX
 931   TEM(I) = TEM(I) + CURMIN(J)*COXY(J,I)
       DO 932 I=1,NY
 932   EY(MY1,I) = COY(I)+TEM(I)
       KY = NY
       CALL LINP(EY,MY,KY,ERY,YY,SU,ZY,ISY)
       DO 933 J=1,NX
       TEM(J) = 0.0
       DO 933 I=1,NY
 933   TEM(J) = TEM(J) + YY(I)*COXY(J,I)
       DO 935 J= MSTAR1,MX1
       ERX(J)=0.0
       DO 935 I = 1, NX
 935   EX(J,I) = 0.0
       DO 934 I=1,NY
 934   EX(MSTAR1,I) = COX(I)+TEM(I)
        LO =NX
       CALL LINP(EX,MSTAR,LO,ERX,X,SU,Z,IS)
       WRITE (6,892)
 892   FORMAT (1H ,///,5X,'NUMBER OF CUTS GENERATED ',//)
       WRITE (6,394) KKK
```

```
 894  FORMAT (10X,I4)
      WRITE(6,895) ZMIN
 895  FORMAT(1H ,///,5X,'GLOBAL MINIMUM ',F12.6)
      WRITE (6,896)
 896  FORMAT (1H ,///,5X,'GLOBAL MINIMIZING POINT',//)
      WRITE (6,897)
 897  FORMAT (5X, 'X VARIABLES',//)
      WRITE(6,898) (X(I),I=1,NX)
 898  FORMAT (5X,8F12.6)
      WRITE(6,899)
 899  FORMAT (1H ,//,5X,'Y VARIABLES',//)
      WRITE (6,898)(YY(I),I=1,NY)
 9924 INST = INST+1
      IF(INST.LT.NUM) GO TO 9923
      END
      SUBROUTINE LINP(A,M,N,D,X,SU,Z,IS)
C      THIS  SUBROUTINE SOLVES A  LP OF THE FORM
C
C      MAX    CX
C      ST     AX LE D
C             X GE 0
C      WHERE D MAY BE GE  OR  LE 0
C      THE LAST ROW OF A CONTAINS THE ELEMENTS C
C      THE REVISED SIMPLEX METHOD WITH ARTIFICIAL VARIABLES IS USED
      DIMENSION A(50,75),B(50,50),CT(75),AT(50),IB(50),
     2INDIC(180),D(50),X(75),Y(170),SU(50),COST(110)
       COMMON B,IB,INDIC,CT,AT
      M1=M+1
      BIGM=-2000
      ZE = 0.0
      NN=N
      DO 1375 I=1,N
      X(I) = 0.0
 1375 COST(I) = A(M1,I)
C      IF ANY RHS IS NEGATIVE,THAT ROW IS MULTIPLIED BY -1,A SURPLUS
C      VARIABLE IS ADDED IN THAT ROW AND -BIGM*THE ROW IS ADDED TO THE
C      LAST ROW ?COSTING OUT THE ARTIFICIAL VARIABLES#
      DO 200 I=1,M
      IF(D(I).GE.0) GO TO 200
      N=N+1
      A(I,N)=1.
      DO 190 J=1,N
      IF(ABS(A(I,J)).LE.0.000005) GO TO 190
      A(I,J)=-A(I,J)
      A(M1,J)=A(M1,J)-BIGM*A(I,J)
  190 CONTINUE
      ZE = ZE -BIGM*D(I)
  200 CONTINUE
C      INITIALIZE
      NT = N + M
      DO 1373 I = 1,NT
      Y(I) = 0.0
 1373 INDIC(I) = 0
      DO 1391 I = 1,M1
      AT(I) = 0.0
      SU(I) = 0.0
      B(M1,I) = 0.0
 1391 IB(I) = 0
```

```
      N1=N+1
C      THE ARRAY INDIC IS +1 FOR A BASIC VARIABLE, -1 FOR A BASIC
C      ARTIFICIAL VARIABLE, AND 0 FOR A NON BASIC VARIABLE
      DO 85 I=1,M
      KK=N+I
      IF (D(I)) 127,123,123
 127  INDIC(KK)=-1
      GO TO 210
 123  INDIC(KK)=1
C      COL N1 OF A CONTAINS THE RHS WITH ALL ELEMENTS .GE.0
 210  A(I,N1) = ABS(D(I))
C     THE ARRAY IB CONTAINS THE INDEX OF THE BASIC VARIABLE FOR EACH ROW
 85   IB(I) = KK

C      THE INITIAL BASIS B AND BINVERSE IS THE IDENTITY MATRIX OF SLACK
C      AND ARTIFICIAL VARIABLES
      DO 59 I=1,M
      DO 59 J=1,M
      IF(I.NE.J) GO TO 70
      B(I,J)=1.
       GO TO 59
   70 B(I,J)=0.
   59 CONTINUE
C      THE COL M1 OF B CONTAINS THE RHS AND B(M1,M1) CONTAINS OBJ. FUN.
      DO 71 I= 1,M
   71 B(I,M1)=A(I,N1)
      B(M1,M1) = ZE
C      THE REL COST FACTORS ARE IN THE ARRAY CT
      DO 72 I=1,N
   72 CT(I)=A(M1,I)
      GO TO 73
   22 CONTINUE
C      COMPUTE REL COST FACTORS
      DO 10 J=1,N
      CT(J)=A(M1,J)
      DO 10 K=1,M
 10   CT(J) = CT(J) + B(M1,K)*A(K,J)
C      REL COST FACTORS FOR BASIC VARIABLES ARE 0
      DO 130 J=1,N
 130  IF(INDIC(J).EQ.1.OR.INDIC(J).EQ.-1) CT(J)=0.
   73 CONTINUE
C      FIND MAX OF REL COST FACTORS IN CM WITH INDEX IN IX-NEW BASIC COL
      CM=CT(1)
      IX=1
      DO 11 J=2,N
      IF(CT(J).LE.CM) GO TO 11
      CM=CT(J)
      IX=J
   11 CONTINUE
C      FIND MAX OF CM AND REL COST FACTORS FOR NONBASIC SLACKS
      DO 12 J=1,M
      IF(B(M1,J).LE.CM) GO TO 12
      CM=B(M1,J)
      IX = N+J
   12 CONTINUE
C      STOPPING RULE
      IF(CM.LE..05) GO TO 1000
      IF(IX.GT.N) GO TO 13
C      THE ARRAY AT IS THE UPDATED ENTERING BASIC COL
```

```
      DO 14 I=1,M
      AT(I)=0.
      DO 14 J=1,M
   14 AT(I)=AT(I)+B(I,J)*A(J,IX)
      AT(M1)=CM
      GO TO 87
   13 IG = IX
      DO 88 J=1,M1
   88 AT(J) = B(J,IX-N)
   87 CONTINUE
      K=0
      DO 15 I=1,M
      IF(AT(I).LE.0.005) GO TO 15
      IF(K.NE.0) GO TO 17
      IR=I
      BM=B(I,M1)/AT(I)
      K=1
      GO TO 15
   17 BZ=B(I,M1)/AT(I)
      IF(BZ-BM) 90,91,15
   90 KZ=0
      BM=BZ
      IR=I
      GO TO 15
   91 KZ=KZ+1

      GO TO 15
   15 CONTINUE
C      TEST FOR UNBOUNDED SOLUTION
      IF(K.EQ.0) GO TO 1001
C      IB(IR) IS THE LEAVING BASIC VARIABLE
      IIIX=IABS(IB(IR))
      INDIC(IIIX)=0
      IIIX=IABS(IX)
      INDIC(IIIX)=1
      IB(IR)=IX
C      COMPUTE NEW B INVERSE
      DO 20 I=1,M1
      IF(I.EQ.IR) GO TO 20
      XY = AT(I)/AT(IR)
      DO 19 J=1,M1
   19 B(I,J)=B(I,J)-XY*B(IR,J)
   20 CONTINUE
      DO 21 I=1,M1
   21 B(IR,I)=B(IR,I)/AT(IR)
      GO TO 22
C      IS ARTIFICIAL VARIABLE BASIC
 1000 DO 401 I = N,KK
      IF (INDIC(I).EQ.-1) GO TO 1017
  401  CONTINUE
      GO TO 1006
 1017 DO 1013 J = 1,M
      IF (IB(J) .EQ.I) GO TO 1014
 1013 CONTINUE
C      IS BASIC ARTIFICIAL VARIABLE %0
 1014 IF (B(J,M1).LE.0.005) GO TO 1006
C      IS = -1 FOR INFEASIBILITY, 1 FOR UNBOUNDEDNESS, 0 OTHERWISE
      IS = -1
      Z=0.0
```

```
      GO TO 5000
 1001 IS = 1
      Z = 500000.0
      GO TO 5000
 1006 CONTINUE
      DO 48 I=1,M
      JKL = IB(I)
 48   Y(JKL) = B(I,M1)
C      THE VECTOR X CONTAINS THE REAL VARIABLES
      DO 49 I=1,NN
 49   X(I) =Y(I)
      JJ=N+1
      JKJ=NN+1
C      THE VECTOR SU CONTAINS SLACK/SURPLUS VARIABLES
      DO 61 I=1,M
      IF (D(I)) 62,63,63
 62   SU(I) = Y(JKJ)
      JKJ=JKJ+1
      JJ=JJ+1
      GO TO 61
   63 SU(I) =Y(JJ)
      JJ=JJ+1
  61  CONTINUE
      IS = 0
      Z =0.0
      DO 57 I = 1,NN
  57  Z = Z-X(I)*COST(I)
 5000 CONTINUE
      RETURN
      END
```

BIBLIOGRAPHY

1. Altman, M., "Bilinear Programming," *Bulletin d' Academie Polonaise des Sciences, Serie des Sciences Math. Astr. et Phys.*, Vol. 19, 741-746 (1968).

2. Balas, E., "Intersection Cuts--A New Type of Cutting Planes for Integer Programming," *Operations Research,* Vol. 19, 19-39 (1971).

3. Balas, E., "Nonconvex Quadratic Programming Via Generalized Polars," Management Science Research Report No. 278, Carnegie-Mellon University GSIA (1972).

4. Balas, E., and Burdet, C. A., "Maximizing a Convex Quadratic Function Subject to Linear Constraints," Management Science Research Report No. 299, Carnegie-Mellon University GSIA (1973).

5. Balinski, M. L., "An Algorithm for Finding All Vertices of a Convex Polyhedral Set," *SIAM Journal on Applied Mathematics,* Vol. 9, 72-88 (1961).

6. Bowman, V. J. and Nemhauser, G. L., "Deep Cuts in Integer Programming," *Opsearch,* Vol. 8, 89-111 (1971).

7. Burdet, C. A., "Polaroids: A New Tool in Nonconvex and in Integer Programming," *Naval Research Logistics Quarterly,* Vol. 20, 13-22 (1973).

8. Burdet, C. A., "Simple Polaroids for Nonconvex Quadratic Programming," Management Sciences Research Report No. 283, Carnegie-Mellon University GSIA (1972).

9. Burdet, C. A., "Generating All the Faces of a Polyhedron," *SIAM Journal on Applied Mathematics,* Vol. 26 (1974) (to appear).

10. Cabot, V. A., and Francis, R. L., "Solving Certain Nonconvex Quadratic Minimization Problems by Ranking the Extreme Points," *Operations Research,* Vol. 18, 82-86 (1970).

11. Candler, W. and Townsley, R. J., "The Maximization of a Quadratic Function of Variables Subject to Linear Inequalities," *Management Science,* Vol. 10, 515-523 (1964).

12. Cooper, L., "Location-Allocation Problems," *Operations Research,* Vol. 11, 331-343 (1963).

13. Cooper, L., "The Transportation-Location Problem," *Operations Research*, Vol. 20, 94-108 (1972).

14. Cooper, L., "Heuristic Methods for Location-Allocation Problems," *SIAM Review*, Vol. 6, 37-52 (1964).

15. Cottle, R. W., and Ferland, J. A., "Matrix Theoretic Criteria for the Quasi-Convexity and Pseudo-Convexity of Quadratic Functions," *Linear Algebra and Its Applications*, Vol. 5, 123-136 (1972).

16. Cottle, R. W., and Mylander, W. C., "Ritter's Cutting Plane Method for Nonconvex Quadratic Programming," in *Integer and Nonlinear Programming* (J. Abadie, ed.), North Holland, 1970.

17. Falk, J. E., and Soland, R. M., "An Algorithm for Separable Non-convex Programming Problems," *Management Science*, Vol. 15, 550-569 (1969).

18. Gallo, G., and Ülkücü, A., "Bilinear Programming: An Exact Algorithm," Operations Research Center Report No. ORC 73-26, University of California, Berkeley, California (1973).

19. Garfinkel, R. S., and Nemhauser, G. L., *Integer Programming*, John Wiley, 1972.

20. Gomory, R. E., "An Algorithm for Integer Solutions to Linear Programs," in *Recent Advances in Mathematical Programming*, R. L. Graves and P. Wolfe, editors, McGraw-Hill, 1963.

21. Grünbaum, B., *Convex Polytopes*, John Wiley, 1967.

22. Hadley, G., *Linear Programming*, Addison-Wesley, 1963.

23. Hillier, F. S., and Lieberman, G. J., *Introduction to Operations Research*, Holden-Day, 1972.

24. Hu, T. C., "Minimizing a Concave Function in a Convex Polytope," Mathematics Research Center Report No. 1011, U. S. Army, Madison, Wisconsin (1969).

25. Kirby, M. J. L., Love, H. R., and Swarup, K., "Extreme Point Mathematical Programming," *Management Science*, Vol. 18, 540-549 (1972).

26. Konno, H., "Bilinear Programming," Parts I and II. Technical Report No. 71-9 and 71-10, Operations Research House, Stanford University (1971).

27. Langley, R. W., "Continuous and Integer Generalized Flow Problems," Unpublished Ph.D. Dissertation, Georgia Institute of Technology.

28. Lasdon, L., *Optimization Theory for Large Systems,* MacMillan, 1970.

29. Mangasarian, O. L., *Nonlinear Programming,* McGraw-Hill, 1969.

30. Martin, G. T., "An Accelerated Euclidean Algorithm for Integer Linear Programming," in *Recent Advances in Mathematical Programming,* R. L. Graves and P. Wolfe, editors, McGraw-Hill, 1963.

31. Martos, B., "The Direct Power of Adjacent Vertex Programming Methods," *Management Science*, Vol. 17, 241-252 (1965).

32. Martos, B., "Subdefinite Matrices and Quadratic Forms," *SIAM Journal on Applied Mathematics,* Vol. 17, 1215-1223 (1969).

33. Moreno, C., "Minimization of Concave Functions Subject to Linear Constraints," Operations Research Center Report ORC 72-3, University of California, Berkeley (1972).

34. Mueller, R. K., "A Method for Solving the Indefinite Quadratic Programming Problem," *Management Science,* Vol. 5, 333-339 (1970).

35. Murty, K., "Solving the Fixed-Charge Problem by Ranking the Extreme Points," *Operations Research,* Vol. 16, 268-279 (1968).

36. Mylander, W. C., "Nonconvex Quadratic Programming by a Modification of Lemke's Method," Report No. RAC-TP-414, Research Analysis Corporation, 1971.

37. Raghavachari, M., "On Connections Between Zero-One Integer Programming and Concave Programming Under Linear Constraints," *Operations Research,* Vol. 17, 680-684 (1969).

38. Ritter, K., "A Method for Solving Maximum-Problems with a Non-concave Quadratic Objective Function," *Z. Wahrscheinlichkeitstheorie verw.,* 4, 340-351 (1966).

39. Shachtman, R., "Generation of the Admissible Boundary of a Convex Polytope," *Operations Research,* Vol. 22, 151-159 (1974).

40. Shachtman, R., Private Communication (May, 1974).

41. Shanno, D. F., and Weil, R. L., "'Linear' Programming with Absolute Value Functionals " *Operations Research,* Vol. 19, 120-124 (1971).

42. Soland, R. M., "Optimal Facility Location with Concave Costs," *Operations Research,* Vol. 22, 373-382 (1974).

43. Srinivasan, V., and Thompson, G. L., "An Operator Theory of Parametric Programming for the Transportation Problem--I,"

Naval Research Logistics Quarterly, Vol. 19, 205-252 (1972).

44. Steinberg, D. I., "The Fixed Charge Problem," *Naval Research Logistics Quarterly,* Vol. 17, 217-235 (1970).

45. Taha, H. A., "Concave Minimization Over a Convex Polyhedron," *Naval Research Logistics Quarterly,* Vol. 20, 533-548 (1973).

46. Tui, H., *Concave Programming Under Linear Constraints* (Russian), Doklady Akademii Nauk SSSR (1964). English Translation in *Soviet Mathematics,* 5, 1437-1440 (1964).

47. Wesolowsky, G. O., and Love, R. F., "The Optimal Location of New Facilities Using Rectangular Distances," *Operations Research,* Vol. 19, 124-129 (1971).

48. Zangwill, W. I., *Nonlinear Programming--A Unified Approach,* Prentice-Hall, 1969.

49. Zwart, P. B., "Global Maximization of a Convex Function with Linear Inequality Constraints," *Operations Research,* Vol. 22, 602-609 (1974).

50. Zwart, P. B., "Nonlinear Programming: Counterexamples to Two Global Optimization Algorithms," *Operations Research,* Vol. 21, 1260-1266 (1973).

51. Zwart, P. B., "Computational Aspects on the Use of Cutting Planes in Global Optimization," *Proceedings 1971 Annual Conference of the Association of Computing Machinery,* 457-465, Chicago (1971).

VITA

Harish Vaish was born October 6, 1940, in Calcutta, India. He was graduated from St. Xavier's Collegiate School, Calcutta, India, in June, 1956. In June, 1958, he passed the Intermediate (Science) examination of Calcutta University. He entered the Indian Institute of Technology, Kharagpur, in July, 1958, from which he graduated in May, 1962, with a B. S. in Mechanical Engineering. From June, 1962, until August, 1969, he worked for the Heavy Machine Building Plant, Ranchi, India, as a mechanical design engineer. In September, 1969, he enrolled in the Department of Industrial Engineering at the University of Tennessee, Knoxville. He completed the requirements for a M. S. in August, 1970, and the degree was awarded in December, 1970. In September, 1970, he enrolled in the Ph. D. program at the School of Industrial and Systems Engineering of the Georgia Institute of Technology. In September, 1972, he joined the faculty at the Georgia Institute of Technology as a part-time Instructor. He completed all the requirements for a Ph. D. in August, 1974, and the degree was awarded in December, 1974. He is currently an Assistant Professor at the California State University, Northridge.